



Estimação de Estados aplicada a um Centro de Telecomando Hidroelétrico

FRANCISCO ILÍDIO MAIA DA ROCHA FERRAZ

julho de 2020

Instituto Superior de Engenharia do Porto
Departamento de Engenharia Electrotécnica
Rua Dr. António Bernardino de Almeida, 431, 4249-015 Porto

Estimação de Estados aplicada a um centro de telecomando hidroelétrico

Mestrado em Engenharia Eletrotécnica e de Computadores - Sistemas Autónomos

Francisco Ilídio Maia da Rocha Ferraz
1130438@isep.ipp.pt

Orientação:
Mário Jorge de Andrade Ferreira Alves, mjf@isep.ipp.pt
Hélder José Sousa Teixeira, helder.teixeira@edp.com

Ano Letivo: 2019-2020

Resumo

Este documento tem como objetivo a apresentação do trabalho desenvolvido ao longo do 2º semestre para a cadeira de Tese / Dissertação, focando-se na projeção e implementação de um modelo de Estimação de Estados, aplicado ao Centro de Telecomando das Centrais Hidroelétricas da EDP Produção. Estes modelos consistem num conjunto de algoritmos direccionados para a previsão de medidas em tempo real, mais concretamente o nível da Cota de Montante, juntamente com uma base de dados histórica de prévios acontecimentos. Primeiramente, captados através dos sensores instalados nas barragens, deve ser obtido um conjunto de parâmetros de entrada predefinidos, relativos ao aproveitamento hidroelétrico, que o sistema utilizará de suporte para produzir um resultado final. Esses dados serão sempre apresentados, em tempo real, no Sistemas de Supervisão e Aquisição de Dados. Visto o Centro de Telecomando Único das Centrais Hidroelétricas ser responsável pela operação e gestão dos aproveitamentos hidroelétricos, é importante que possíveis anomalias nas leituras realizadas pelos sensores, provenientes das centrais sejam imediatamente detetadas, de modo a que os operadores consigam atuar em conformidade. Muitas das atividades desta equipa passam pelo operação remota dos aproveitamentos e a garantia que a exploração é realizada cumprindo as normas existentes, daí à existência de um sistema de monitorização deste género permite que estes se concentrem nas suas atividades do dia-a-dia, pois qualquer evento fora do comum será detetado pelo modelo de estimação.

O foco deste trabalho foi o estudo das implementações de *Machine Learning* já existentes no mercado, bem como a perceção de quais as possíveis tecnologias integrantes neste tipo de soluções. Seguidamente, com base na análise realizada, o destaque passou para a preparação do modelo de estimação de estados, desenvolvida em linguagem *Python*, utilizando a ferramenta *Tensorflow* para a criação da rede neuronal. O modelo tem como principal objetivo a validação da evolução das cotas, em função de um conjunto específico de critérios, presentes em cada uma das albufeiras, possibilitando a análise e deteção de erros no nível da cota de montante, ou outras variáveis registadas nas instalações, como o nível da cota de jusante ou o caudal afluente, turbinado e descarregado da barragem. A realização de um estudo aprofundado dos dados históricos das centrais foi essencial para o estimador produzir resultados mais eficientes e fidedignos.

Palavras Chave:

Estimador de Estados, Centro de Telecomando, Aproveitamento Hidroelétrico, Machine Learning, SCADA, previsão, estimação, base de dados, tempo real, otimização

Abstract

This document goal is to present the thesis project developed throughout the 2nd semester, focused on studying and implementing a State Estimation model, applied to EDP's hydroelectric remote control center. These models consist on a set of algorithms, aimed at making real time value predictions, using an historical database of previous events. The set of input parameters, that the model will use to produce a final output, will be obtained from direct readings from the Supervisory and Data Acquisition System. Being responsible for the management and operation of the hydroelectric powerplants, the Hydroelectric Remote Control Center needs to have a way of monitoring and detecting possible anomalies in the data coming from the plants, so that operators are able to act accordingly. Many of the activities of the teams involve remote conduction of the facilities and guaranteeing that the work is made in compliance with the existing standards. The existence of a monitoring system of this kind allows them to focus on their day to day tasks, as any event out of the ordinary will be captured model.

The main purpose of this work is to study Machine Learning applications already that exist in the current market, as well as to understand what kind of technologies are part of this type of products, in order to gain a better understanding on how these solutions can be developed. Based on the previous analysis, the highlight of the project moves to the preparation of the state estimation model, using the Python language and the Tensorflow framework to create a neural network. The goal of this model is to validate the current water storage levels, according to a specific set of pre-established criterias, providing a deep analysis, and error detections, of the measures registered in the power plants. Carrying out an in-depth study of the historical data of the plants will be essential for the estimator to produce more efficient and reliable results.

Keywords:

State Estimator, Remote Control Center, Hydroelectric Power Plant, Machine Learning, SCADA, prediction, estimate, database, real time, optimization

Conteúdo

Conteúdo	i
Lista de Figuras	v
Lista de Tabelas	vii
Lista de Equações	ix
Lista de Algoritmos	xi
Glossário	xiii
Agradecimentos	xv
1 Introdução	1
1.1 Contextualização	1
1.2 Motivação	2
1.3 Identificação do Problema e Objetivos	3
1.4 Arquitetura Global	4
1.5 Calendarização	5
1.6 Contribuições	6
1.7 Organização da Dissertação	6
2 Operação dos Aproveitamentos Hidroelétricos	9
2.1 Aproveitamentos Hidroelétricos	9
2.1.1 Evolução da exploração dos aproveitamentos hidroelétricos	9
2.1.1.1 Operação local	9
2.1.1.2 Operação distanciada	10
2.2 Gestão das albufeiras em situações de risco	10
2.2.1 Cheias	10
2.2.2 Secas	11
2.3 Aquisição de dados	11
2.3.1 Medição de cotas	11
2.4 Estimação de Estados	11
2.4.1 Requisitos do sistema	12
2.4.2 Estimação de cotas	12

2.4.3	Funcionamento em tempo real	13
2.4.4	Previsão de cotas	13
3	Avaliação de metodologias e tecnologias de <i>Machine Learning</i>	15
3.1	Metodologias	15
3.1.1	<i>Supervised Learning</i>	15
3.1.2	<i>Unsupervised Learning</i>	16
3.1.3	<i>Reinforcement Learning</i>	17
3.1.4	<i>Deep Learning</i>	17
3.2	Redes Neurais	17
3.2.1	Redes Perceptrons	18
3.2.2	Redes Neurais Recorrentes	18
3.2.3	Redes Neurais Convolucionais	19
3.2.3.1	<i>Faster R-CNN</i>	20
3.2.4	<i>You Only Look Once</i>	20
3.2.5	<i>Long Short-Term Memory</i>	21
3.3	<i>Frameworks</i>	22
3.3.1	TensorFlow	22
3.3.1.1	Keras	22
3.3.2	Apache MXNet	22
3.3.3	Caffe	22
3.3.4	Theano	22
3.4	Cenários de aplicação	23
3.4.1	Reconhecimento de objetos	23
3.4.2	Previsões	23
3.4.3	Recomendação de produtos e serviços	23
3.4.4	Automação e controle	23
3.5	Treinar o modelo	24
3.6	Conclusões	24
4	Análise e correlação das variáveis presentes nos sistemas hidroelétricos	27
4.1	Estudo do sistema	27
4.1.1	Indicadores principais	27
4.1.2	Barragem piloto	29
4.1.3	Indicadores adicionais	30
4.1.4	Correlação dos parâmetros	31
4.2	Conclusões	34
5	Desenvolvimento e Integração	35
5.1	Ambiente de Desenvolvimento	35
5.2	Data	36
5.3	Desenvolvimento	37
5.3.1	Inicializações	38
5.3.2	Repartição do conjunto de dados	38
5.3.3	Implementação da rede neuronal	39
5.3.3.1	Criação da rede neuronal para o modelo de estimação de estados	42
5.3.4	Fluxograma dos algoritmos	45

5.3.5	Previsão de cotas	47
6	Resultados e Validações	49
6.1	Testes de validação	49
6.2	Desempenho do modelo	54
6.3	Previsões futuras para além do <i>dataset</i>	56
6.4	Conclusões	59
7	Conclusões	61
7.1	Contribuições	61
7.2	Trabalho futuro	62
	Bibliografia	65

Lista de Figuras

1.1	Diagrama de Blocos: Receção dos dados das Centrais Hidroeléctricas	4
1.2	Diagrama de blocos: Funcionamento em tempo real do Estimador de Estados	5
1.3	Diagrama de blocos: Desenvolvimento do modelo de Estimação de Estados . .	5
1.4	Calendarização do projeto	6
2.1	Ilustração da exploração das centrais hídricas [1]	12
3.1	Exemplo de <i>Machine Learning</i> supervisionado [2]	16
3.2	Processo de classificação de dados não designados [3]	16
3.3	Exemplo de uma rede neuronal com 4 camadas [3]	17
3.4	Classificação de Rede Perceptron de uma única camada [4]	18
3.5	<i>Loop</i> das Redes Neuronais Recorrente [5]	18
3.6	Matriz RGB das Redes Neuronais Convolucionais [6]	19
3.7	Redes Neuronais Convolucionais: <i>Fully Connected Layer</i> [6]	20
3.8	Modo de classificação do método YOLO [7]	21
3.9	Modo de processamento de um modelo LSTM [8]	21
3.10	Classificação de objetos com <i>Machine Learning</i> [9]	23
3.11	Imagem ilustrativa do funcionamento de um conjunto de veículos autónomos [10]	24
4.1	Ilustração dos tipos de caudais [1]	28
4.2	Esquema transversal de uma barragem [11]	28
4.3	Vista de jusante da Barragem da Bouçã [12]	29
4.4	Vista superior da Barragem do Cabril [1]	30
4.5	Sequência do Rio Zêzere [1]	31
4.6	Cenário 1 da Variação horária do nível da Cota de Montante, em metros, da Barragem da Bouçã	32
4.7	Cenário 1 da Variação horária dos Caudais, em metros cúbicos por segundo, da Barragem da Bouçã	32
4.8	Cenário 2 da Variação horária do nível da Cota de Montante, em metros, da Barragem da Bouçã	32
4.9	Cenário 2 da Variação horária dos Caudais, em metros cúbicos por segundo, da Barragem da Bouçã	33
4.10	Cenário 1 da Variação horária do nível da Cota de Jusante, em metros, da Barragem da Bouçã	33

4.11	Cenário 2 da Variação horária do nível da Cota de Jusante, em metros, da Barragem da Bouçã	34
5.1	Demonstração do IDE Spyder [13]	36
5.2	Representação da divisão das series temporárias na rede neuronal.	41
5.3	Exemplo do processamento das sequências.	41
5.4	Dependência a longo prazos nas Recurrent Neural Network's [14]	42
5.5	Módulo de repetição de uma RNN padrão. [14]	42
5.6	Comparação de uma Sigmóide em relação a uma camada <i>tanh</i> [15]	43
5.7	O módulo de repetição presente numa rede LSTM [14]	43
5.8	Camada sigmóide [14]	44
5.9	Fluxograma da criação do modelo de estimação de estados.	46
5.10	Fluxograma da execução do modelo de estimação de estados.	46
6.1	Previsões realizadas com 10 <i>epochs</i> e 120 <i>steps</i>	50
6.2	Previsões realizadas com 100 <i>epochs</i> e 60 <i>steps</i>	50
6.3	Previsões realizadas com 100 <i>epochs</i> e 6 <i>steps</i>	51
6.4	Previsão horária do nível da Cota de Montante, em metros, realizada com 200 <i>epochs</i> e 24 <i>steps</i>	52
6.5	Previsão horária do nível da Cota de Montante, em metros, realizada com 500 <i>epochs</i> e 6 <i>steps</i>	52
6.6	Desvios existentes na previsão com 500 <i>epochs</i> e 6 <i>steps</i>	53
6.7	Previsão horária do nível da Cota de Montante, em metros, realizada com 500 <i>epochs</i> e 3 <i>steps</i>	53
6.8	Desvio, em metros, entre a Cota de Montante Real e a Prevista	54
6.9	Variação, em metros, da Cota de Montante da Bouçã entre os dias 21/12/2019 e 22/12/2019	54
6.10	Curva de perda do modelo de Estimação de Estados	55
6.11	Curva de validação da perda do modelo de Estimação de Estados	55
6.12	Previsão futura além do conjunto de dados existente (1)	56
6.13	Relação, em metros, entre a Cota de Montante Real e Prevista de 02/03/2020	57
6.14	Previsão futura além do conjunto de dados existente (2)	57
6.15	Análise da relação, em metros, entre a Cota de Montante Real e Prevista de 09/03/2019 (1)	58
6.16	Análise da relação, em metros, entre a Cota de Montante Real e Prevista de 09/03/2019 (2)	58

Lista de Tabelas

5.1	Configuração base do <i>dataset</i> referente à Barragem da Bouçã	37
5.2	Nova escala do <i>dataset</i> da Barragem da Bouçã	40
6.1	Resultados das previsões futuras além do <i>dataset</i> (1)	56
6.2	Resultados das previsões futuras além do <i>dataset</i> (2)	59

Lista de Equações

4.1	Calculo do Caudal Natural da Bouçã	30
-----	--	----

Lista de Algoritmos

1	Query de recolha de dados da Barragem da Bouçã	36
2	Query de recolha de dados da Barragem do Cabril	37
3	Primeiro conjunto de <i>import's</i>	38
4	Segundo conjunto de <i>import's</i>	38
5	Leitura e distribuição dos dados	39
6	Transformação do <i>dataset</i>	40
7	Criação do modelo de estimação de estados	44
8	Treino do modelo de estimação de estados	45
9	Inclusão de um período futuro no <i>dataset</i>	47
10	Prevendo para além do conjunto de dados	47

Glossário

- AGO** Gestão da Operação. xv
- APA** Agência Portuguesa do Ambiente. 2
- API** Application Programming Interface. 22, 24
- CO2** Dióxido de carbono. 9
- CPU** Central Processing Unit. 22
- CTCH** Centro de Telecomando Único das Centrais Hidroelétricas. iii, 1–4, 7, 11–13, 53, 61, 62
- DOH** Direção de Otimização e Gestão de Ativos Hídricos. xv
- EDP** Energias de Portugal. iii, v, xv, 1, 2, 4, 6, 10, 24, 62
- GPU** Graphics Processing Unit. 22, 41
- IDE** Integrated Development Environment. vi, 24, 35, 36
- LSTM** Long Short-Term Memory. v, vi, 21, 22, 25, 42–44, 47, 61
- MSE** Mean Squared Error. 45
- NPA** Nível Pleno de Armazenamento. 32
- O&M** Organização, sistemas e métodos. 4
- R-CNN** Region Convolutional Neural Network. 20
- RGB** Red Green Blue. 19
- RNA** Redes Neuronais Artificiais. 17
- RNN** Recurrent Neural Network. vi, 41, 42
- RPN** Region Proposal Network. 20

SCADA Sistemas de Supervisão e Aquisição de Dados. 3, 4, 62

YOLO You Only Look Once. v, 20, 21

Agradecimentos

Desejo desde logo expressar a minha gratidão para com os meus orientadores Professor Mário Alves e Eng. Hélder Teixeira, pela sua disponibilidade, a sua dedicação, o seu apoio e toda a ajuda na partilha de conhecimentos e informação durante o decorrer deste projeto.

Igualmente agradeço ao Eng. Vítor Silva, por ter tornado possível a concretização deste projeto, e por todo o suporte e acompanhamento prestado ao longo desta fase, fazendo-me adquirir um maior conhecimento do mundo das hídricas.

Agradeço também a toda a equipa da Direção de Otimização e Gestão de Ativos Hídricos: Gestão da Operação, da EDP Produção, por toda a disponibilidade e apoio prestado, para que este projeto fosse um sucesso.

Aos meus Pais, agradeço toda a preocupação, toda a motivação e por sempre acreditarem nas minhas capacidades.

À minha avó, Maria Augusta, agradeço por ter sido sempre o meu maior apoio durante todo o meu percurso académico e profissional, torcendo sempre por mim em todas as ocasiões.

Por fim, à minha namorada, Cátia Neves, agradeço por toda a força e motivação que deu ao longo dos últimos anos.

Capítulo 1

Introdução

Neste capítulo será apresentado o tema a ser abordado ao longo da Tese de mestrado, o que foi desenvolvido em volta deste tópico, e o respetivo planeamento e execução do trabalho ao longo do semestre.

1.1 Contextualização

O conceito de estimação de estados traduz-se na geração de valores de referência para um conjunto de valores de entrada, provenientes de um dado sistema. Tipicamente, o processo de desenvolvimento de um modelo de estimação pode ser dividido em múltiplas etapas, passando-se inicialmente sempre por uma fase de estudo do sistema, de modo a perceber como é que todas as suas variáveis se correlacionam entre si. Entendendo essa relação, deve ser realizada uma reflexão sobre quais os resultados a ser produzidos pelo estimador. Isso leva a que os parâmetros com maior impacto no resultado final sejam, logicamente, aqueles utilizados como entrada para o sistema.

A partir do histórico de dados armazenados na base de dados do Centro de Telecomando Único das Centrais Hidroelétricas, é possível perceber como é que a variação de certos parâmetros alteram consoante determinadas circunstâncias. Colocando um algoritmo de *Machine Learning* a verificar esses acontecimentos, podemos concluir com um maior grau de certeza, que de facto certos padrões existem, podendo ser utilizados para prever, ou validar, determinados acontecimentos.

A atividade de produção de energia elétrica, a partir da utilização de águas fluviais, remonta a meados do século XIX, sendo que em Portugal só começou a eclodir na última década desse século.

A história da EDP, no domínio hidroelétrico começa a desenhar-se no início da década de 50, através da construção dos primeiros aproveitamentos hidroelétricos. Findada a sua construção, iniciou-se uma nova etapa, a Operação, realizada localmente e com recurso a um número muito significativo de recursos humanos distribuídos pelas mais diversas funções (sala de comando, sala de máquinas, barragem, ...).

Nas décadas seguintes assistiu-se a grandes avanços tecnológicos e eis que no ano de 1972 se dá o primeiro grande passo para o que é hoje a operação à distância (Teleco-

mando): foi possível realizar o controlo remoto da central de Vilarinho das Furnas a partir da sala de comando da central da Caniçada.

As alterações de paradigma conjugadas com o amadurecimento e fiabilidade das tecnologias, permitiu no início do século XXI (ano 2000) concentrar a operação remota de todas as centrais num único local, o CTCH (Centro de Telecomando de Centrais Hidroelétricas), situação que se regista aos dias de hoje.

A criação do Centro de Telecomando possibilitou uma melhor gestão dos ativos hídricos, unificando e uniformizando a informação de exploração e controlo, e claro, otimizando a utilização dos recursos, isto tudo garantindo uma operação segura e contínua dos grupos geradores.

A última década foi inédita, superando até então os anos de ouro da hidroeletricidade em Portugal, anos 50 e 60, reescrevendo-se uma nova história com a instalação de mais 2300 MW. A EDP Produção é responsável pela operação de 62 Centrais hidroelétricas, com uma potência total de aproximadamente 7 GW.

A evolução natural dos sistemas informáticos possibilitou um aumento de capacidade e de velocidade de processamento de informação permitindo um incremento exponencial do número de variáveis a monitorizar. Isto, conjugado com a pressão da otimização da exploração dos ativos, induz a necessidade de criar ferramentas capazes de tratar e analisar um conjunto de informação em tempo-real.

A tecnologia avança e novas formas e abordagens emergem, possibilitando a melhoria contínua dos processos, sendo esta Tese um claro exemplo. Este projeto visou dotar o telecomando hidroelétrico da EDP de uma ferramenta de alerta e de auxílio na tomada de decisão, através da aplicação de ferramentas de *Machine Learning* e *Data Analytics*.

O algoritmo de estimação será portanto desenvolvido tendo em consideração todas estas constantes e calibrado à medida que este vai sendo testado, de modo ajustar os seus resultados ao aproveitamento hidroelétrico em questão. Reunindo então os dados históricos dos aproveitamentos, estabelecendo a ligação, em tempo real, do conjunto de *inputs* a receber, aplicando uma rede neuronal sobre esses valores de entrada e finalmente, utilizando o algoritmo de geração de resultados, teremos o modelo criado, capaz de monitorizar e estimar resultados com um enorme grau de confiança.

1.2 Motivação

Cada vez mais existe uma maior apreensão, por parte das entidades reguladoras, como a Agência Portuguesa do Ambiente, para a garantia de um correto funcionamento dos ativos hídricos, de forma a manter um elevado grau de eficiência e ao mesmo tempo de eficácia. Isto leva a que haja uma certa preocupação para encontrar novos critérios de decisão à operação. A gestão das afluências de modo a otimizar a produção e o cumprimento de obrigações contratuais, como a garantia dos caudais ecológicos e o lançamento de caudais para abastecimento de água, são presentemente atividades de enorme responsabilidade.

O Centro de Telecomando Único das Centrais Hidroelétricas tem como missão garantir a operação e condução remota dos aproveitamentos hidroelétricos, sendo algumas das suas responsabilidades:

- coordenação e regularização dos níveis das albufeiras, cumprindo as normas de exploração e as obrigações legais vigentes;
- estabelecimento de contactos operacionais com as autoridades e organismos oficiais;
- colaborar nas atividades de análise de incidentes ou ocorrências de exploração, no seu tratamento sistemático e na implementação de ações de melhoria;
- coordenar o relacionamento operacional com a unidade de *trading*, de forma a atingir elevados níveis de desempenho e maximizando os proveitos.

Para que ocorra uma resposta assertiva no que é a génese e função do CTCH, a operação, há um fator fundamental e crucial no processo: a qualidade dos dados. Essa qualidade é determinada a partir da forma como os dados são gerados, tratados e armazenados e é essencial no processo operacional, sendo importante criar mecanismos de análise com o intuito de detetar possíveis inconsistências ou anomalias. Uma boa qualidade dos dados possibilita uma maior eficácia e eficiência na gestão dos ativos, e garantirá a credibilidade da empresa perante os seus *stakeholder*.

Partindo deste princípio e tendo em linha de conta as valências técnicas e conhecimentos adquiridos durante o percurso académico, criou-se uma janela de oportunidade para produzir algo que permita suportar e auxiliar a operação do CTCH, isto é, transformar dados em informação credível, que possa ser utilizada no processo de decisão e inclusive suportar o planeamento estratégico de curto-prazo.

Por todos estes motivos, o *Estimador de Estados* tornou-se uma necessidade a ser implementada, de modo a servir de auxílio ao trabalho realizado pelas equipas operacionais. O módulo de estimação efetuará uma constante validação das medidas/sinais do sistema permitindo a deteção de incoerências. Essa verificação deve ser realizada com base num histórico de passados acontecimentos, valores de referência e correlação de dados preexistentes. Com base neste conjunto de parâmetros, sempre que o modelo detetar uma falha, será gerada, em tempo real, uma notificação para o operador, agilizando o processo de validação ou reparação das medidas.

1.3 Identificação do Problema e Objetivos

As sinalizações de estados e medidas que constituem a base de dados do SCADA são recolhidas localmente através de equipamentos muito distintos, quer do ponto de vista tecnológico quer da sua vida útil restante, e na sua grande maioria expostos a condições de funcionamento adversas. Esta situação pode conduzir a falhas nos equipamentos, nas medições e nas comunicações, que podem não ser detetadas no instante em que ocorrem, induzindo os operadores, e o próprio sistema, em erro.

Nos dias de hoje a tecnologia apresenta-se como o principal motor de desenvolvimento e progresso. Na indústria será, com toda a certeza, a chave do sucesso para a competitividade estratégica das empresas e para o seu desenvolvimento económico, na medida em que trará aporte de valor aos mais diversos produtos ou serviços. A transformação digital será a base para aumentar a competitividade tecnológica e económica das empresas, que sentem a necessidade de se reinventarem, através da concessão e desenvolvimento de processos de inovação tecnológica. A inovação tecnológica está na ordem do dia, nos

mais diversos fóruns, media, meios académicos e empresariais, e a EDP não é exceção. A EDP quer liderar a transição energética e, portanto, está a apostar em desenvolver-se na área da digitalização, de forma a que possa ser referência a nível mundial, reconhecendo que as soluções digitais trarão uma ampla gama de benefícios.

A EDP Produção, e em particular, a área de operação - Centro de Telecomando das Centrais Hídricas (CTCH), irá reforçar a aposta em avanços tecnológicos e dotar a operação dos ativos hídricos de ferramentas capazes de permitir um aumento da flexibilidade e da fiabilidade. Assim, a presente tese é subordinada ao tema “Estimador de Estados como ferramenta de otimização de O&M”, cujo objetivo será o desenvolvimento de um algoritmo que permita a estimação de variáveis fundamentais no processo de telecomando dos aproveitamentos hidroelétricos da EDP Produção, dotando-a de uma capacidade de resposta mais incisiva e célere em caso de falha.

1.4 Arquitetura Global

O procedimento para o qual o envio e receção dos dados é realizado, decorre da seguinte forma:

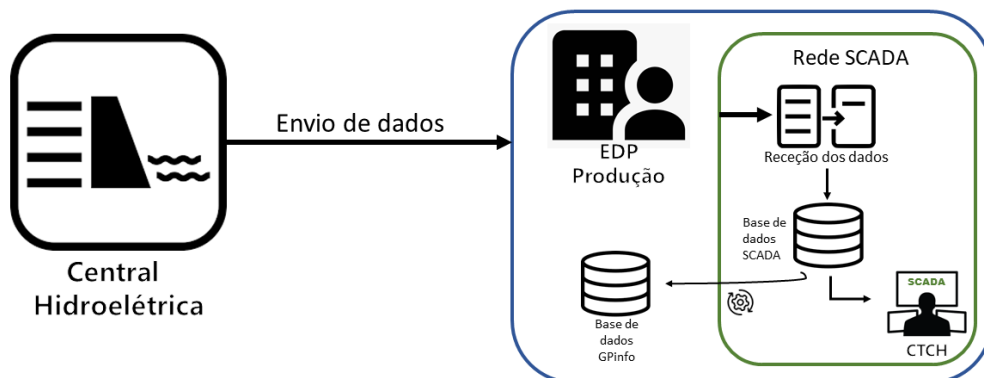


Figura 1.1: Diagrama de Blocos: Receção dos dados das Centrais Hidroelétricas

De forma a proteger a integridade do funcionamento do CTCH, a rede onde se encontra o SCADA, está isolada das restantes, evitando assim que um possível ataque informático à rede EDP afete a operação das centrais hidroelétricas, garantindo assim a segurança de pessoas e de bens.

Após os dados serem armazenados, quer na base de dados GPInfo, quer na base de dados SCADA, torna-se viável o tratamento e validação dessa informação.

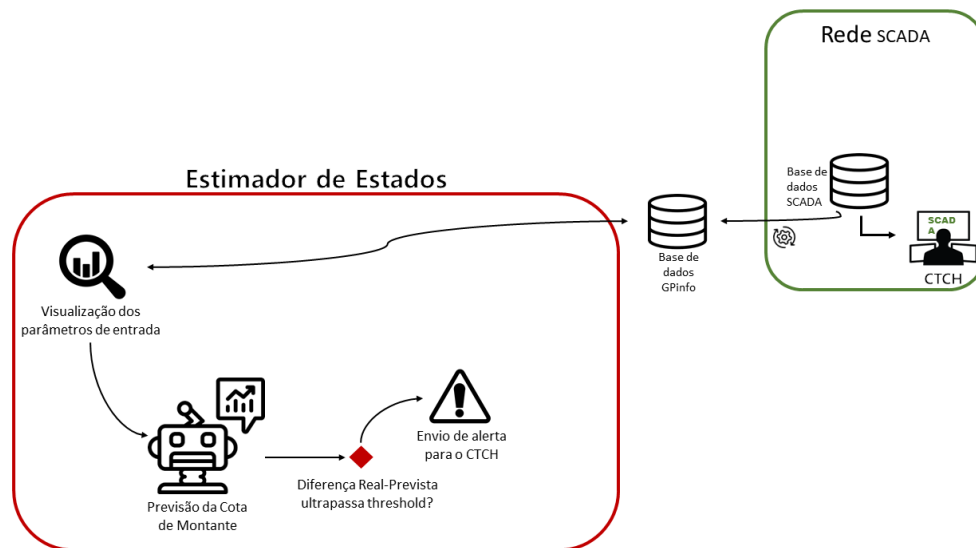


Figura 1.2: Diagrama de blocos: Funcionamento em tempo real do Estimador de Estados

A preparação, construção e calibração do modelo de estimação de estados passou pelo seguinte conjunto de passos:

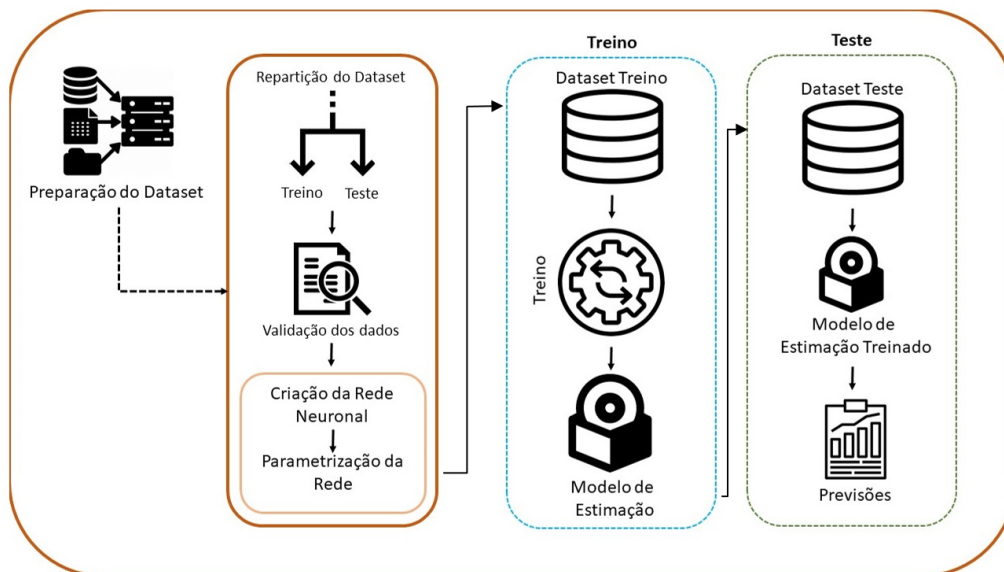


Figura 1.3: Diagrama de blocos: Desenvolvimento do modelo de Estimação de Estados

1.5 Calendarização

Sendo o estudo das possíveis tecnologias a ser implementadas no modelo de estimação, juntamente com o conhecimento de quais os dados a ser analisados, de forma a poder obter os melhores resultados possíveis, o grande desafio por trás deste projeto, a sua

prossecação conduziu à calendarização apresentada na seguinte figura, que apresenta os múltiplos pontos referidos na secção 1.3.

De acordo com as tarefas descritas, apresenta-se a seguinte calendarização tecnológica:

	Feb	Mar	Abr	Mai	Jun	Jul
Planeamento do Projeto	█					
Estudo das Tecnologias	█	█				
Pesquisa sobre Possíveis Aplicações		█				
Desenvolvimento do Modelo		█	█			
Validação e Testes			█	█		
Elaboração do Relatório Final	█	█	█	█	█	
Revisão do Relatório e Preparação da Defesa						█

Figura 1.4: Calendarização do projeto

1.6 Contribuições

As principais contribuições deste projeto, face às necessidades da EDP Produção foram:

- Estudo e análise das metodologias de *Machine Learning* existentes;
- Pesquisa das diversas *frameworks* no mercado, apropriadas para implementações de aprendizagem máquina;
- Estudo das correlações existentes, entre as diversas variáveis presentes nas barragens hidroelétricas;
- Construção de um *dataset* de treino e de teste, com as variáveis da barragem selecionada, para o modelo de estimação de estados;
- Desenvolvimento e calibração do modelo de estimação de estados, preparado para prever o nível da Cota de Montante da Barragem da Bouçã;
- Realização de múltiplos testes de *performance* ao modelo de estimação;

1.7 Organização da Dissertação

No Capítulo 1 é introduzido o tema selecionado para a cadeira de Tese / Dissertação, juntamente com a motivação e os objetivos deste projeto. No Capítulo seguinte, 2, é realizado um aprofundamento ao tópico relativo à operação e gestão dos aproveitamentos hidroelétricos da EDP Produção, e também, apresentadas possíveis situações de risco que possam vir a ocorrer nas albufeiras. No Capítulo 3, é realizada uma apresentação às metodologias de *Machine Learning* existentes, como também os diferentes tipos de redes neuronais estudadas, para a criação do modelo de estimação de estados. Do mesmo modo, são identificadas as *frameworks* existentes no mercado, para o desenvolvimento e integração deste tipo de soluções. No Capítulo 4 é selecionada a barragem piloto para

este projeto e realizada uma análise das variáveis presentes nos sistemas hidroelétricos, de forma a perceber como estas se relacionam, e como o modelo de estimação pode aprender e realizar previsões, através do seu histórico de eventos. No Capítulo 5, são apresentados todos os passos realizados de modo a produzir, com sucesso, uma rede neuronal robusta, capaz de estimar, e conseqüentemente validar, em tempo real, os parâmetros enviados para o CTCH. No Capítulo 6, são demonstrados os testes realizados ao modelo de estimação, de modo a calibrar e assim tornar, o seu *output* o mais eficaz possível. No Capítulo 7, são reunidas as principais conclusões, e por fim, a dissertação é finalizada com as referencias bibliográficas utilizadas no trabalho desenvolvido.

Capítulo 2

Operação dos Aproveitamentos Hidroelétricos

Neste capítulo será apresentada uma breve descrição, relativa aos aproveitamentos hidroelétricos e os seus respectivos modos de operação. Também é realizada uma análise, e posterior apresentação, dos requisitos necessários para o desenvolvimento de um modelo de estimação de estados, aplicado a essas infraestruturas, juntamente com conjunto de tecnologias analisadas e implementadas para o seu desenvolvimento.

2.1 Aproveitamentos Hidroelétricos

Desde a década de 1940, que a utilização de aproveitamentos hidroelétricos, para geração de energia, iniciou em Portugal. Estas infraestruturas contêm um papel essencial na gestão das redes elétricas pois proporcionam, de uma forma rápida e moderadamente confiável, a produção de energia através de uma fonte renovável. O seu potencial produtivo está diretamente relacionado com o território em que o aproveitamento se encontra localizado [16], daí existir sempre um vasto estudo prévio à construção das barragens. Cada barragem provoca um enorme impacto ecológico, sendo que por esse motivo, essa perda ambiental deve-se justificar a longo prazo. A diminuição do uso de combustíveis fósseis, poluentes líquidos ou gasosos, e a ausência de emissões de CO₂, são algumas das vantagens dos aproveitamentos hidroelétricos [16].

2.1.1 Evolução da exploração dos aproveitamentos hidroelétricos

2.1.1.1 Operação local

À medida de que foram sendo construídos novos aproveitamentos hidroelétricos, sentiu-se um enorme aumento no número de elementos distribuídos pelas diversas funções de gestão dos ativos, sendo criadas novas funções direcionadas para operar na sala de comando, na sala das máquinas, e também na própria barragem. “*Na sala de comando, para além da informação dos grupos geradores, existiam também equipamentos de proteções e comando assim como informação sobre a área hidrológica, nomeadamente cotas, sendo a vigilância, comando e diálogo com o despacho feitos a partir deste local pelos operadores*

aí existentes... Na sala de máquinas estavam residentes duas pessoas cuja missão era efetuar o arranque dos grupos geradores até atingirem a velocidade de sincronismo assim como, de forma inversa, procederem à paragem completa dos geradores, após o pessoal da sala de comando retirar o grupo da rede... Existiam ainda guardas de instalações e nas barragens distantes das respetivas centrais, podendo ser chamados em qualquer momento desde que as condições assim o exigiam (cheias, etc.), apresentado uma média de trabalhadores por grupo gerador elevada” [17] [18]. Devido à contínua evolução das metodologias de operação, uma maior necessidade para a automatização foi requerida, até que no final dos anos oitenta surgiram as primeiras centrais com operação à distância, em que o comando, e todas as atividades à sua volta, se encontravam centralizadas por computador.

2.1.1.2 Operação distanciada

No ano de 1972, foi instalado o primeiro dispositivo de “teleoperação”, que fez com que a partir da Caniçada, fosse possível comandar remotamente a Central de Vilarinho de Furnas, sendo que desde aí, novas automatizações de “teleoperação” foram implementadas para as restantes centrais. No ano 2000, foi então decidido telecomandar-se todos os aproveitamentos hidroelétricos, a partir do centro em Bagúste – Peso da Régua, centralizando-se assim esta atividade. Presentemente, esse comando passou a ser realizado no edifício da EDP, situado na Boavista. A contínua evolução das tecnologias de exploração existentes levou a uma grande diminuição no número de colaboradores por grupo gerador. Em 1991, esse número equivalia a 4.7 trabalhadores, sendo que nos dias de hoje, a média ronda os 0.15. A criação de um Centro de Telecomando centralizado foi uma enorme mais valia na gestão e operação dos recursos hídricos, possibilitando uma maior segurança no trabalho, juntamente com um aumento da rapidez e fiabilidade das atividades.

2.2 Gestão das albufeiras em situações de risco

A garantia de uma correta utilização dos recursos hídricos é um tema que constantemente acompanha o negócio da EDP Produção, tanto que a construção das infraestruturas deve ser profundamente analisada, pois é o que permite a minimização dos impactos causados quer pelo excesso ou escassez de chuva. Ao longo do ano, os trabalhadores devem modificar por completo a forma como operam, dependendo do período em que se encontram, visto que os níveis de retenção de água dependem das atuais necessidades e dos seus fins [19].

2.2.1 Cheias

A exploração das albufeiras em situações de cheias, isto é, em períodos de elevadas afluências, pode ser uma prática bastante dificultosa, pois requer bastante experiência e conhecimento, de forma a cumprir-se escrupulosamente as regras de exploração das albufeiras, com o objetivo de salvaguarda de pessoas e de bens, a montante e a jusante, garantindo a segurança das instalações. Relacionado às albufeiras, estão sempre associados os níveis máximos e mínimos que a água pode atingir, bem como quais as regras e normas que se deve seguir durante as situações que obrigam a realização de descarregamentos. Cada vez que é efetuado um descarregamento, é necessário ter em consideração

quais a normas estabelecidas para o lançamento de caudais, em função do afluente à albufeira e a variação horária do caudal lançado total [19].

2.2.2 Secas

A exploração das albufeiras em situações de seca é uma atividade que implica um grande sentido de gestão e economização do volume de água disponível nas barragens. Nestas ocasiões, a produção de energia não é o mais prioritário pois deve ser assegurada a existência de água para o fornecimento público, seguido das restantes atividades fundamentais para o dia a dia das populações. As situações de seca podem até mesmo exigir que ocorra uma total restrição da utilização das centrais de modo a evitar o esvaziamento da albufeira, sendo que a EDP introduz sempre medidas de contenção de utilização, para casos em que os níveis de cota das albufeiras estejam abaixo dos valores médios estabelecidos [19].

2.3 Aquisição de dados

A constante monitorização e gestão dos ativos é realizada consoante o conjunto de parâmetros que chegam ao CTCH em tempo real. Sendo estes referentes ao atual estado do aproveitamento, dados como a cota de montante e de jusante, ou o caudal turbinado, afluente e descarregado, são variáveis fundamentais de apoio à tomada de decisão dos operadores. Estes dados são captados por sensores instalados nas barragens que se encontram em permanente comunicação com o centro de telecomando.

2.3.1 Medição de cotas

A cota de montante, é usada como referência para saber o atual nível de água, armazenada no depósito da barragem. Este valor é lido através de sensores instalados na barragem, designados por limnímetros, que medem as variações do nível da superfície da água. De modo a automatizar o processo de recolha desta informação, evitando assim que tenha de ser um operador a realizar a leitura manual, as centrais são compostas por sistemas que medem os sensores, digitalizam a medida e enviam a informação para o centro de telecomando.

A cota de jusante é utilizada para saber o nível de água a jusante da barragem, sendo um dos parâmetros técnicos utilizados para a validação da possibilidade de funcionamento dos grupos geradores.

2.4 Estimação de Estados

O processo de estimação de estados consiste em fornecer ao *end user*, o grau de certeza que existe relativamente a um resultado obtido. Esse grau de certeza é sempre determinado consoante um dado conjunto de parâmetros de entrada, sendo que o estimador processa essas medidas de modo a prever qual seria o *output* espectável, o que por si tem um grande impacto na exploração que é realizada. Para além da validação de medidas, o modelo de estimação permite que o utilizador ganhe uma nova perceção de como as variáveis existentes no sistema se correlacionam entre si, sendo que por vezes certas conclusões não são evidentes a “olho nu”. A análise e deteção de padrões ocultos

nos dados históricos dos aproveitamentos hidroelétricos permite uma melhor compressão de como os elementos interagem entre si, possibilitando otimizações nas metodologias de exploração presentemente aplicadas.

2.4.1 Requisitos do sistema

O modo de processamento, e posterior obtenção, dos resultados desejados, difere sempre do cenário de execução, daí a que o primeiro passo a alinhar passa sempre por definir qual a aplicação desejada. Para que o modelo seja robusto e apresente resultados fidedignos, será sempre necessário um conjunto de dados históricos, previamente validados, de forma a garantir que as conclusões foram alcançadas segundo informações confiáveis. Por este motivo, é essencial realizar um pré-diagnóstico de toda a informação a ser recebida pelo estimador, validando que os dados que se encontram na base de dados apresentem-se coerentes e fiáveis, durante a fase de tomada de decisões dos algoritmos. De modo a captar padrões e correlações entre os dados existentes no sistema, o foco seria de aplicar uma metodologia de *Supervised* ou *Unsupervised Machine Learning*, colocando uma rede neuronal a treinar sobre o *dataset* existente na base de dados do CTCH. Para tal, outro trabalho a realizar trata-se de identificar que tipo redes neuronais poderiam ser integradas para este cenário de aplicação. Decidida qual a abordagem de *Machine Learning* a aplicar e que tipo de rede neuronal utilizar, um grande conjunto de diferentes treinos do modelo deve ser realizado, a modo a verificar, com um elevado grau de certeza, qual dos métodos apresentou os melhores resultados. Garantindo que o modelo apresenta resultados robustos, proporciona-se assim ao operador, um software de auxílio na tomada de decisão, bastante conveniente para cenários extremos, i.e. cheias ou secas, ou até mesmo receção de dados anómalos.

2.4.2 Estimação de cotas

Considerando o peso de cada um dos parâmetros envolvidos na operação diária dos aproveitamentos hidroelétricos, concluiu-se que para este projeto piloto, a estimação da cota de montante seria um bom caso para comprovar a eficiência do modelo. Diretamente relacionado a este valor encontra-se o caudal lançado e o caudal afluente das barragens:

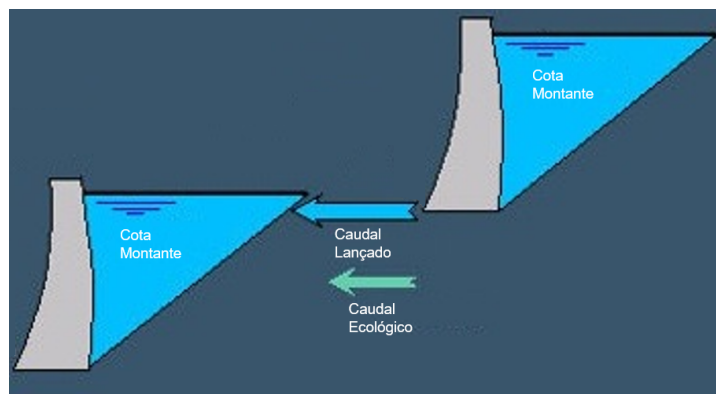


Figura 2.1: Ilustração da exploração das centrais hídricas [1]

O caudal lançado consiste no somatório do caudal turbinado, caudal descarregado

e caudal ecológico, por outras palavras, é a quantidade total de água que é transferida de montante para jusante. Já o caudal afluente representa o conjunto total de água que chega à barragem, sendo este o caudal lançado da barragem anterior, caso esta exista, mais o caudal natural do ambiente em que este se encontra envolvido.

Visto que o caudal natural equivale à quantidade de água que chega à barragem, proveniente de fontes externas, não é possível medir este valor em concreto, logo para efeitos de estimação, apenas será considerado o caudal turbinado e descarregado, juntamente com o caudal afluente.

2.4.3 Funcionamento em tempo real

Existindo um processo de estimação a trabalhar em sincronismo com as equipas operacionais, garante-se que estas conseguem realizar as suas funções de uma forma mais segura. Sendo fornecido ao modelo, uma base de dados confiável, com todos os eventos que foram sucedendo ao longo dos anos, este consegue prever, com um dado grau de certeza, determinados comportamentos padrão que têm acontecido ao longo dos anos. Esta previsão é a chave para começar a moldar o resultado final visto o estimador conter uma base por onde se sustentar e assim ajustar o *output* consoante as variáveis de entrada. Com esses estados é então possível determinar outras variáveis fundamentais para a análise e monitorização do sistema [20]. Além da estimação de valores, o modelo adequa-se para a deteção de inconsistências nos parâmetros registados, visto que podem ocorrer falhas nas leituras por parte dos equipamentos. Identificada uma anomalia, este pode enviar um alerta às equipas para que uma intervenção ocorra de imediato, sem que esta falha de leitura cause atrasos ou lapsos na operação dos aproveitamentos.

2.4.4 Previsão de cotas

Para além da aplicação do modelo de estimação para validar se os parâmetros recebidos no CTCH são fidedignos, este ao ser desenvolvido integrando tecnologias de *Machine Learning*, podemos utiliza-lo para determinar futuras previsões, com base no *dataset* histórico em que este foi treinado. O grau de certeza da previsão, logicamente será sempre menor que o da estimação, no entanto, não impede que seja aplicado como ferramenta de suporte para gerir futuras subidas ou descidas do nível da cota de montante.

Capítulo 3

Avaliação de metodologias e tecnologias de Machine Learning

Com a contínua evolução do Machine Learning, começam a aparecer cada vez mais frameworks de suporte para essa área, contendo um vasto conjunto de bibliotecas e estruturas, que facilitam bastante a integração deste tipo de projetos. Empresas como a Amazon, Facebook, Google e Microsoft encontram-se constantemente a trabalhar para desenvolver novos aplicativos que incorporem estas tecnologias [21], o que não deixa de ser um assunto complexo pois muitas vezes, o facto de uma biblioteca funcionar para um determinado cenário não implica que funcione para outro.

3.1 Metodologias

Dentro do ramo de *Machine Learning*, existem diferentes tipos de abordagens que podem ser aplicadas para melhorar a precisão dos modelos de previsão. Tudo depende da natureza do problema em questão, do tipo de dados a serem processados e do seu respetivo volume de informação.

3.1.1 Supervised Learning

Pelo fato de possuímos previamente os parâmetros necessários, juntamente com as suas respetivas designações, de modo a realizar a previsão da Cota de Montante, o *Supervised Learning* será o método utilizado no desenvolvimento do modelo de Estimação de Estados.

Esta metodologia começa com um conjunto preestabelecido de dados, juntamente com um prévio conhecimento de como estes se encontram classificados. Esse conjunto, conhecido como "*training data*", contém sempre uma, ou mais, entradas e uma única saída, também conhecida por sinal de supervisão. Através de uma otimização iterativa da função objectivo, os algoritmos de *Supervised Learning* criam uma função, que pode ser usada para prever o *output* associado a novas entradas, resultando numa função que permite determinar corretamente novas saídas, para casos que não faziam parte dos dados de treino [22].

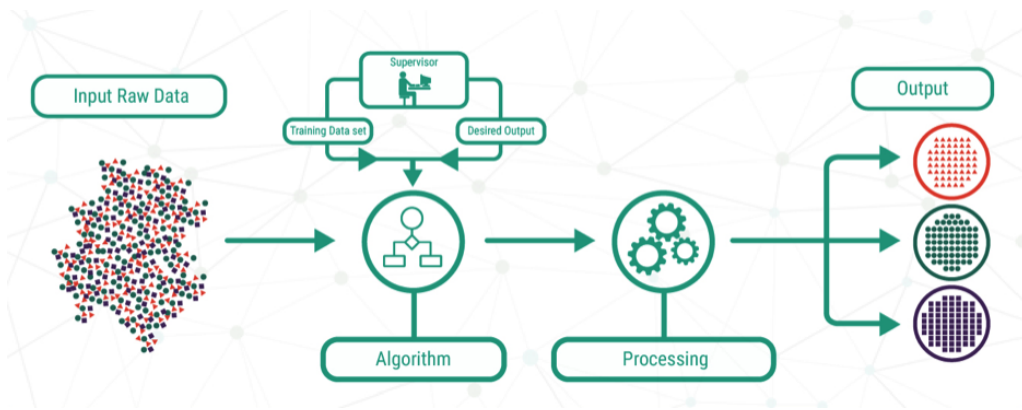


Figura 3.1: Exemplo de *Machine Learning* supervisionado [2]

O maior cenário de aplicação para este tipo de metodologias, destina-se à produção de modelos de detecção de padrões, que possam depois ser aplicados para diversos tipos de processos analíticos.

3.1.2 Unsupervised Learning

O *Unsupervised Learning* é aplicado em cenários para os quais o problema contém uma grande quantidade de dados não designados. Isto implica que antes seja necessário primeiramente entender o seu significado, o que requer a utilização de algoritmos de classificação que se alimentem dos padrões encontrados, de modo a realizar a respectiva designação [3]. Esta abordagem conduz um processo iterativo, analisando os dados existentes sem que seja necessário qualquer tipo de intervenção humana.

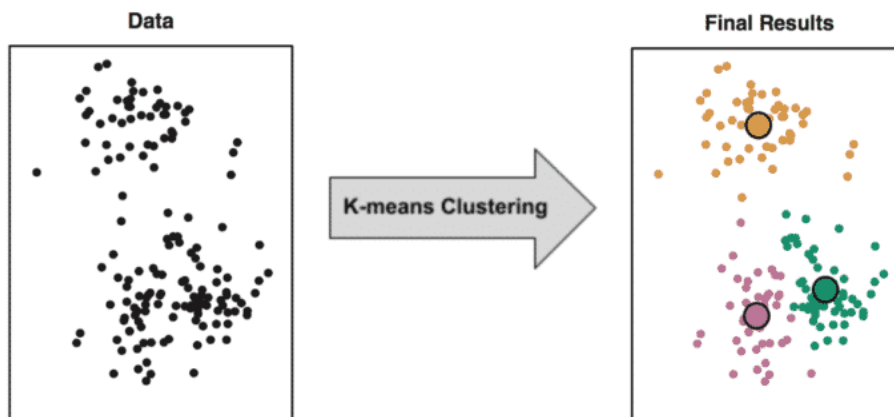


Figura 3.2: Processo de classificação de dados não designados [3]

Os algoritmos de *Unsupervised Learning* identificam pontos comuns no *dataset* existente e reagem com base na presença, ou ausência, desses mesmos pontos em cada novo dado. O algoritmo aplicado na figura ilustrativa da metodologia é o *K-means Cluste-*

ring, cuja função é identificar grupos, ou seja, *clusters* de dados que partilhem valores semelhantes. O cenário em questão identificou com sucesso três *clusters* diferentes [3].

3.1.3 Reinforcement Learning

O *Reinforcement Learning* consiste numa metodologia que aplica modelos de aprendizagem comportamental. Os algoritmos destes modelos recebem um *feedback* referente à análise realizada, sendo esse depois utilizado para gerar um resultado mais eficaz na próxima execução. Ao contrário das restantes metodologias, este método aprende consoante uma abordagem tentativa e erro, o que leva a que uma sequência de decisões bem sucedidas sejam reforçadas pois fizeram com que o sistema resolva-se, da melhor forma, o problema em questão.

3.1.4 Deep Learning

O *Deep Learning* consiste num método específico de *Machine Learning* que incorpora múltiplas redes neuronais em camadas sucessivas, de forma a aprender com os dados apresentados. Esta implementação assemelha-se bastante ao comportamento do cérebro humano, consequentemente preparando os computadores para lidar com problemas mal definidos e abstractos.

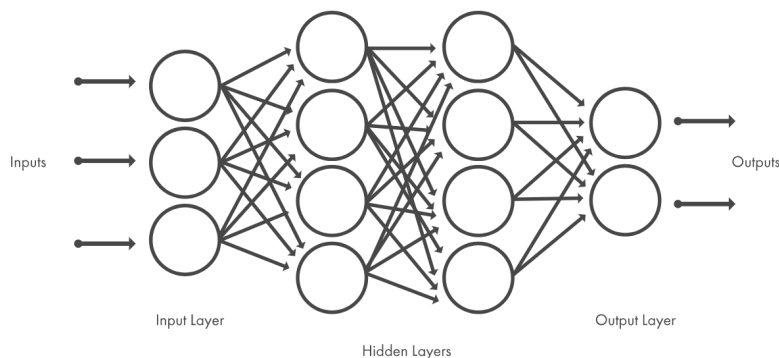


Figura 3.3: Exemplo de uma rede neuronal com 4 camadas [3]

Este tipo de abordagem, apesar de mais complexa que as restantes, demonstra-se muito benéfica quando se pretende detetar padrões em conjuntos de dados não designados, podendo levar ao desenvolvimento de aplicações de reconhecimento facial e de fala, ou visão, computacional.

3.2 Redes Neuronais

As Redes Neuronais Artificiais, ou RNA's, consistem em modelos computacionais semelhantes ao sistema nervoso central, sendo capazes de realizar raciocínios de modo a tirar certas conclusões ou até mesmo reconhecer padrões. Devido ao facto destas redes serem bastante robustas e flexíveis, tornam-se uma grande mais valia na área da classificação de dados, redução de ruído e previsão de resultados [23].

3.2.1 Redes Perceptrons

As Redes Perceptron consistem em algoritmos destinados a classificações binárias, isto é, prevêem se os dados de entrada pertencem a uma determinada categoria de interesse. Estas redes separam duas categorias, traçando uma linha reta entre elas, de modo a classifica-las consoante o peso das variáveis existentes [24].

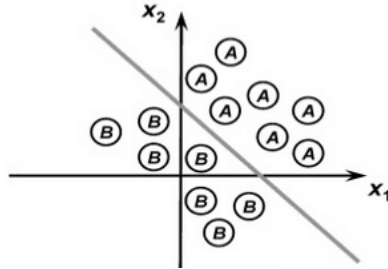


Figura 3.4: Classificação de Rede Perceptron de uma única camada [4]

As redes compostas por mais que um perceptron denominam-se de *Multilayer Perceptrons*. Estas são compostas por uma camada de entrada, uma camada de saída, e entre estas duas, um conjunto variável de camadas ocultas que realizam o processamento. Este método aplica algoritmos de retropropagação do erro, em que cada camada da rede tem uma função específica, fazendo com que a camada de saída construa um resultado final com base nas regras estabelecidas em cada camada. Estas redes são frequentemente aplicadas para problemas de *Supervised Learning*, em que treinam com um conjunto de pares entrada e aprendem a correlacionar e modular as dependências entre eles.

3.2.2 Redes Neurais Recorrentes

As Redes Neurais Recorrentes são aplicadas para o processamento de dados sequenciais, series temporais e linguagem natural, contendo um *loop* de *feedback's* constantes.

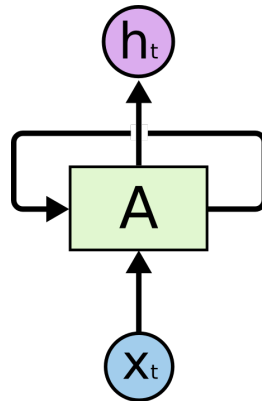


Figura 3.5: Loop das Redes Neurais Recorrente [5]

A saída ($n-1$) é aplicada de volta no sistema de forma a incorporar o estado do modelo no cálculo do resultado seguinte (n), sendo a mesma metodologia aplicada para

a seguinte etapa, produzindo assim o *loop* de *feedback's*. O modelo recorrente inclui um estado "oculto" que determina a classificação atual que posteriormente é combinada com os dados de entrada do novo passo, para produzir um novo estado 'oculto' e a sua respetiva classificação.

3.2.3 Redes Neurais Convolucionais

As Redes Neurais Convolucionais são as principais redes a ser utilizadas para classificar imagens e agrupa-las por similaridade. Detecção de objectos e reconhecimento facial, são algumas das áreas em que estas redes são bastante úteis.

Para tal, o processo de classificação recebe como *input* uma imagem, processa-a e classifica-a dentro de uma determinada categoria, através da visualização dessa mesma imagem como se tratasse de uma matriz de pixels Red Green Blue [6].

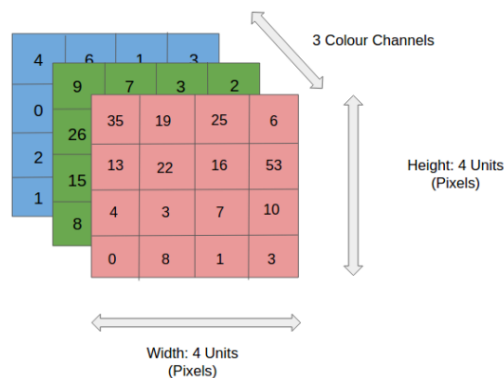


Figura 3.6: Matriz RGB das Redes Neurais Convolucionais [6]

O grau de certeza do *output* gerado por esta metodologia depende sempre da qualidade da resolução da imagem.

Para que estes modelos possam ser treinados e testados, cada *input* vai passar por um conjunto de camadas de convolução, usadas para trabalhar sobre o *blur* e a nitidez da imagem, e detetar os seus limites, juntamente com outros elementos que possam ser relevantes na classificação. Após realizado esse tratamento, as imagens passam por um processo de *Pooling* que permite reduzir o número de parâmetros existentes, sendo posteriormente encaminhadas para um conjunto de camadas totalmente conetadas (*Fully Connected Layer*), de modo a classificar o objeto com valores de 0 a 1.

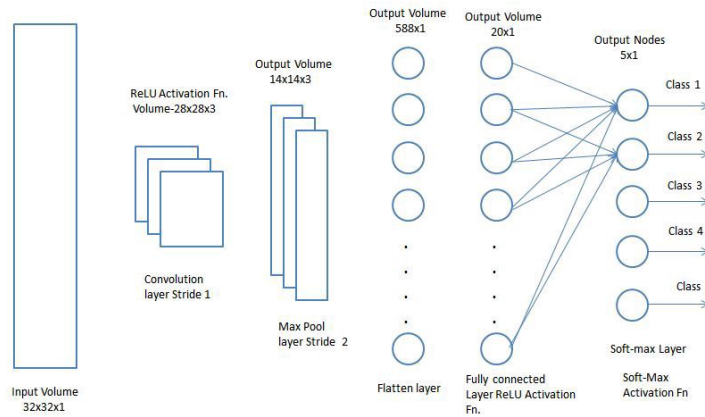


Figura 3.7: Redes Neurais Convolucionais: *Fully Connected Layer* [6]

3.2.3.1 Faster R-CNN

No *Faster R-CNN* ambas as tarefas responsáveis por gerar a proposta da região e a detecção dos objetos, são realizadas pelas mesmas redes convolucionais. Este método possui duas redes, o RPN que irá classificar as regiões existentes de interesse ao sistema em questão, e outra rede que com base nessa proposta, vai detetar os objetos desejados.

A grande diferença desta abordagem é que realiza uma pesquisa selectiva para gerar as propostas referidas.

3.2.4 You Only Look Once

O método *You Only Look Once* é um sistema de detecção de objetos em tempo real, que se demonstra ser extremamente rápido e preciso. Ao contrário das abordagens de localização referidas anteriormente, o seu algoritmo de detecção aplica uma única rede neural à imagem completa. Essa rede, no momento que corre analisa a imagem inteira, resultando com que as suas previsões sejam geradas tendo em consideração o contexto global da imagem. No final esta é dividida em regiões, dimensionando caixas delimitadores para cada objeto detetado, e calculando as respectivas probabilidades de cada região existente [7].

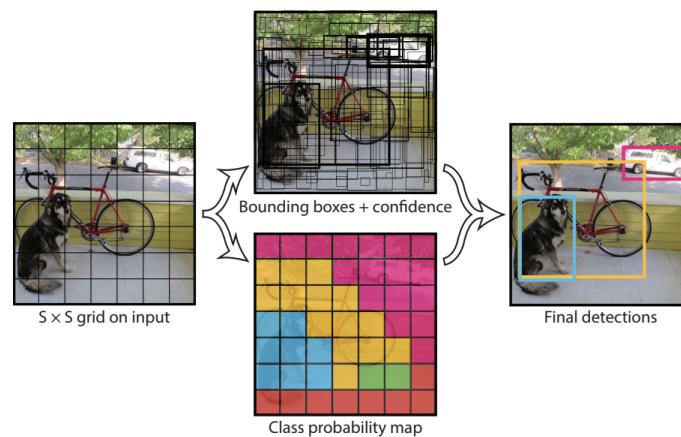


Figura 3.8: Modo de classificação do método YOLO [7]

3.2.5 Long Short-Term Memory

O *Long Short-Term Memory*, abreviado de LSTM, foi apresentado como uma solução para um dos problemas presente nas Redes Neurais Recorrentes. Este problema é denominado de *vanishing gradient* e refere que à medida que mais camadas, que utilizem determinadas funções de ativação, vão sendo adicionadas às redes neurais, os gradientes da função de perda vão-se aproximando de zero, o que dificulta bastante o treino da rede. O LSTM ajuda a preservar esse erro o que permite que as redes recorrentes continuem aprender durante um maior conjunto de passos [25].

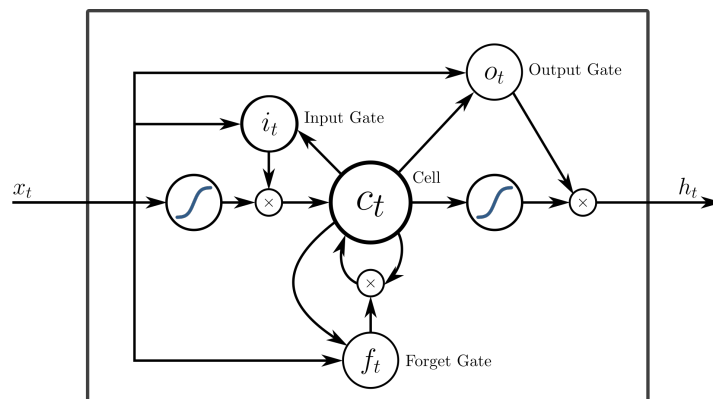


Figura 3.9: Modo de processamento de um modelo LSTM [8]

Esta solução armazena dentro de uma célula fechada, informações fora do fluxo normal da rede recorrente, que podem depois ser escritas ou lidas, como se fossem dados na memória de um computador. A célula toma decisões, através de canais analógicos, sobre o que armazenar, e quando deve permitir realizar as leituras, gravação ou a exclusão dos dados.

3.3 Frameworks

De modo a prestar suporte ao desenvolvimento de modelos de *Machine Learning*, diversos *frameworks* foram criados, facilitando a implementação e adaptação da tecnologia às necessidades em causa.

3.3.1 TensorFlow

O TensorFlow é uma biblioteca *open-source* direcionada a implementações de *Machine Learning*, aplicável a uma ampla variedade de tarefas. Existem inúmeras funcionalidades embudadas nesta *framework* que possibilitam a resolução de problemas mais complexos, a partir do uso de redes neurais artificiais [26].

Projetos como o desenvolvimento de carros autónomos, reconhecimento facial, diagnóstico e cura antecipada de doenças, recomendação de produtos aos clientes, ou até mesmo, a previsão das variações das ações na bolsa de valores, são alguns exemplos das possíveis aplicações que hoje em dia já se encontram produzidas recorrendo ao uso do TensorFlow. Esta biblioteca permite a criação e o treino de redes neurais, de forma a detetar e decifrar padrões e correlações no conjunto de dados a serem analisados.

3.3.1.1 Keras

O Keras é uma API do TensorFlow, escrita em *Python*, para criar e treinar modelos de *Deep Learning*. Esta foi desenvolvida inicialmente para rapidamente testar redes neurais que fossem bastante profundas, no entanto, presentemente um dos seus principais focos consiste na facilidade que existe em manobrar a ferramenta.

3.3.2 Apache MXNet

À semelhança do TensorFlow, o Apache MXNet consiste num *software open-source* de *Deep Learning*, utilizado para criar, de uma forma rápida e escalável, redes neurais. O MXNet inclui uma *interface* denominada de "Gluon", que facilita imenso o trabalho de quem se encontra a iniciar desenvolvimentos na área do *Machine Learning*. O Gluon permite criar regressões lineares, redes convolucionais e LSTM's, utilizadas posteriormente para tarefas de reconhecimento e deteção de parâmetros [27].

3.3.3 Caffe

O Caffe consiste numa *framework* de *Deep Learning* desenvolvida pela Berkeley AI Research e outros colaboradores da comunidade existente. Apresentando uma arquitetura inovadora, esta *framework* possibilita alternar o uso do CPU e da GPU, para que se consiga treinar uma máquina e de seguida implementar o sistema em *clusters* ou outros dispositivos [28].

3.3.4 Theano

O Theano é uma biblioteca que permite definir, otimizar e eficientemente avaliar expressões matemáticas com matrizes multi-dimensionais [29]. Contém *features* para controlar o uso da GPU, de diferenciação simbólica e também de otimização da velocidade durante análises mais complexas. O seu compilador encontra-se direcionado para

manipular e avaliar expressões matemáticas, em que os cálculos se encontram expressos usando uma sintaxe do tipo *NumPy*.

3.4 Cenários de aplicação

Presentemente existe um grande interesse na utilização de técnicas de *Machine Learning* para inúmeros cenários. Tudo que possa servir de apoio a determinadas decisões, ou até mesmo descobrir e resolver certas incógnitas, torna-se um bom caso de aplicação para esta tecnologia.

3.4.1 Reconhecimento de objetos

Um dos cenários de aplicação mais usuais consiste no reconhecimento de objetos em imagens. Esta técnica pode ser utilizada tanto para deteção de rostos como para reconhecimento de padrões. Para isso é necessário primeiro reunir um *dataset* robusto, com múltiplas imagens dos objetos desejados e treinar as redes neuronais para os captar.

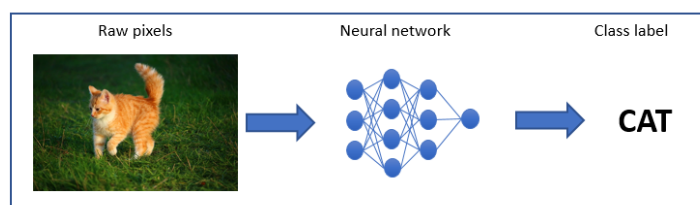


Figura 3.10: Classificação de objetos com *Machine Learning* [9]

3.4.2 Previsões

A partir do uso de *Machine Learning*, é possível detetar determinados padrões nos dados a analisar, o que permite a geração de previsões com base nos históricos existentes. Este tipo de aplicações tende a ser realizada em cenários mais presumíveis, como previsões meteorológicas ou de vendas, no entanto, apesar de não apresentar tanta robustez, esta prática pode também ser empregue para dados mais inconsistentes, como na previsão da variação da bolsa de valores para um dado período de tempo.

3.4.3 Recomendação de produtos e serviços

Este tipo de aplicações são das mais simples e diretas, sendo utilizadas por comerciantes para recomendar artigos que possam vir a ser interessantes para um dado nicho de clientes, com base no seu histórico de compras e pesquisas. Correlacionando informações retidas em base de dados, é possível gerar um *output* que seja cativante para o cliente em questão.

3.4.4 Automação e controlo

O *Machine Learning* é utilizado para uma variedade de sistemas de controlo de robôs e veículos autónomos, existindo presentemente diversos conteúdos *online*, de algoritmos de controlo para este tipo de aplicações. Visto nos dias de hoje estes objetos encontrarem-se

equipados com sensores e atuadores, é possível criar sistemas robustos que possibilitem o controlo autónomo.

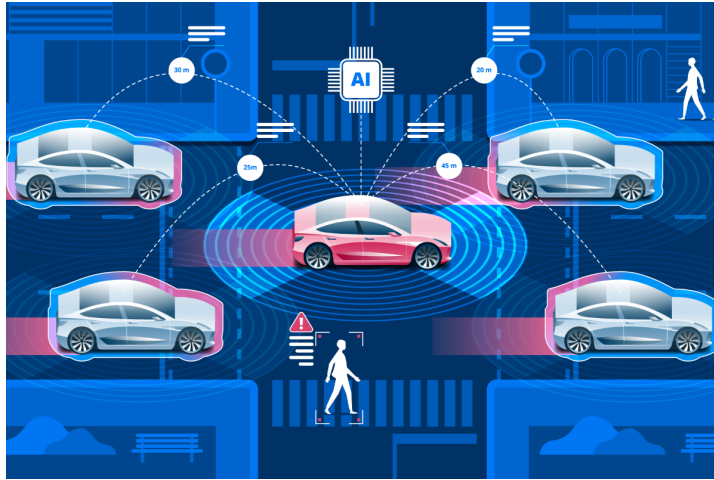


Figura 3.11: Imagem ilustrativa do funcionamento de um conjunto de veículos autónomos [10]

3.5 Treinar o modelo

O *dataset* utilizado para construir o modelo resulta de um conjunto de dados presentes na base de dados GPinfo, contendo registos desde 1 de Janeiro de 2000. Esta base de dados consiste no repositório onde os dados da EDP, respetivos à exploração hídrica, se encontram armazenados.

Prévio à fase de desenvolvimento do modelo, será selecionado um conjunto de variáveis, diretamente relacionadas com a cota de montante das barragens. Esse *dataset* deve passar por múltiplos testes, recorrendo ao uso de um *dataset* de testes, distinto do que foi usado para treinar o modelo. À medida que os resultados são gerados, o sistema vai sendo reajustado, de modo a melhor refletir o seu propósito.

3.6 Conclusões

No que diz respeito ao desenvolvimento de sistemas, considerados "inteligentes", para a análise e validação de dados, o *Machine Learning* é sem dúvida das estratégias mais flexíveis e abrangentes no mercado existente. O facto de existir uma grande quantidade de informação *online*, torna a sua integração de menor complexidade, permitindo ao utilizador ter a liberdade de escolher que IDE, plataforma e linguagem de programação prefere aplicar.

Entre as tecnologias avaliadas, tendo em consideração todos os seus aspectos, simultaneamente com o propósito do projeto em questão, concluiu-se que o Tensorflow, juntamente com a API Keras, seria a *framework* mais indicada. O Tensorflow reúne todos os aspetos necessários para integrar a solução desejada, existindo também imenso conteúdo *on-line* de suporte a este tema.

Manipulando o conjunto de dados reunido, será construída uma Rede Neuronal Recorrente, treinada por uma arquitetura LSTM, pelo facto desta possibilitar a implementação de múltiplas camadas de *feedback*. Através de um método de *continuous value prediction*, o *output* da atual camada serve de *input* para a seguinte. As redes LSTM são indicadas para trabalhar com séries temporais, de modo a realizar classificações, processamentos ou previsões, dos dados em questão.

Capítulo 4

Análise e correlação das variáveis presentes nos sistemas hidroelétricos

Nesta secção será apresentado um estudo das variáveis que mais influenciam a atividade das barragens. De modo a produzir um modelo eficiente com resultados eficazes, reagindo a possíveis variações drásticas do nível da cota de montante, é necessário que o estimador seja treinado com os parâmetros mais significativos. Além dessa tarefa, procede-se à decisão de escolher qual a central/barragem a ser utilizada ao longo do projeto piloto, tendo sempre em consideração que a estimação dos valores em questão, requer uma grande base de dados histórica.

4.1 Estudo do sistema

Ao longo do tempo de vida útil de uma central hidroelétrica, muitas são as características que afetam a sua operação. No que toca à gestão dos ativos hídricos, a cota de montante de uma barragem pode ser definida como o atual nível da água, em relação ao nível médio do mar. Esta variável indica qual o volume de água disponível em cada aproveitamento hidroelétrico possível de convertida em energia.

4.1.1 Indicadores principais

Designa-se por Caudal Afluente a quantidade de água (m^3/s) que chega a uma barragem. Esse caudal é resultado do Caudal Natural dessa mesma barragem e do Caudal Lançado da anterior. O Caudal Natural consiste na quantidade de água que chega, resultante de factores externos, como a chuva ou cursos de água que encaminhem água para o rio. Já o Caudal Lançado representa a soma do Caudal Turbinado, Caudal Descarregado e Caudal Ecológico que é debitado pela barragem.

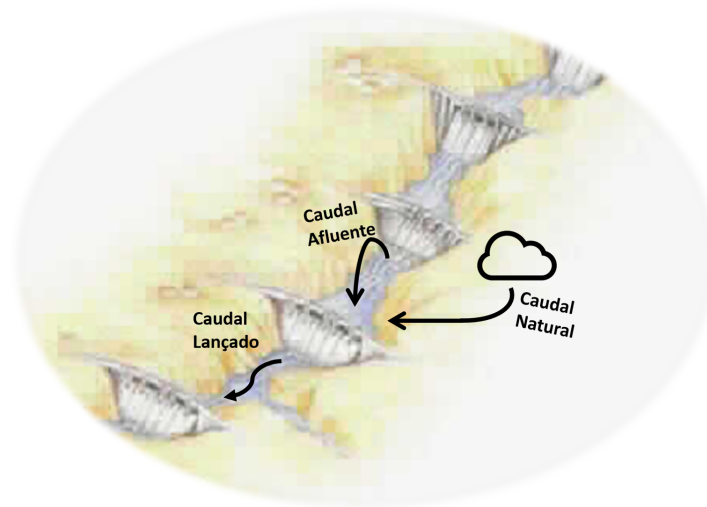


Figura 4.1: Ilustração dos tipos de caudais [1]

O Caudal Turbinado consiste no caudal que passa pelo sistema de turbinas, gerando desse modo a energia necessária para apoiar o negócio.

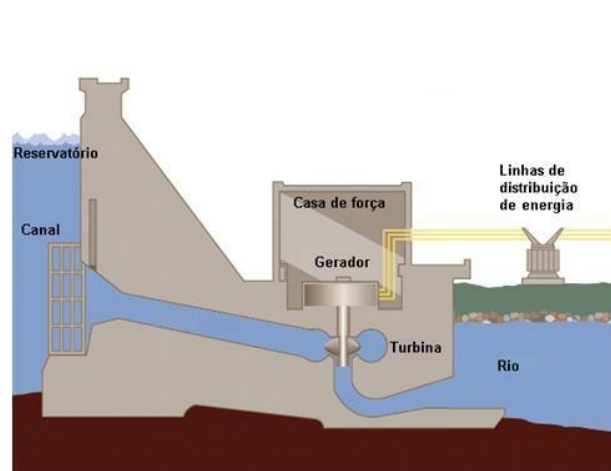


Figura 4.2: Esquema transversal de uma barragem [11]

À semelhança do Caudal Turbinado, o Descarregado também representa uma dada quantidade de água que é lançada pelo descarregador, sendo libertada com o intuito de fazer com que o excedente de água existente seja escoado, evitando assim danos a montante da barragem e o seu colapso. Finalmente, o Caudal Ecológico consiste no caudal mínimo a libertar no curso de água, que permita assegurar a conservação e proteção do ecossistema [30].

Concluindo-se que de facto os caudais de entrada e saída da barragem serão um factor fundamental para a estimação da atual Cota de Montante, decidiu-se treinar o modelo com os dados históricos do Caudal Turbinado, Caudal Afluente e Caudal Descarregado

integrados no *dataset*, sendo o Caudal Ecológico descartado pelo facto dos valores em causa não terem impacto na modelização.

4.1.2 Barragem piloto

Foram aplicados um conjunto de pré-requisitos de forma a facilitar a escolha da central que melhor se ajustaria a este projeto. De modo a comprovar a efetividade desta implementação, um dos pontos a ter em consideração é a existência de um histórico de dados robusto, registando o funcionamento habitual da central, eventos anómalos e situações extraordinárias que tenham ocorrido na albufeira. Para além dessa questão, é preciso garantir que para o piloto, a central selecionada não apresenta demasiadas condicionantes externas, pelo que nesse tipo de cenários esses factores devem ser contabilizados ao longo da fase de treino e de previsão.

Tendo em conta o conjunto de requisitos definidos, chegou-se à conclusão que a barragem da Bouçã seria um bom caso de teste para aplicar o modelo de estimação de estados. A barragem foi inaugurada em 1955, existindo registos de dados históricos prévios ao ano 2000.



Figura 4.3: Vista de jusante da Barragem da Bouçã [12]

Outro factor relevante na escolha consistiu que, além do seu Caudal Natural, a água que lhe chega provém apenas da barragem do Cabril.



Figura 4.4: Vista superior da Barragem do Cabril [1]

4.1.3 Indicadores adicionais

Um conjunto adicional de parâmetros secundários pode ser acrescentado ao *dataset* de treino, assim consolidando os resultados produzidos. Para além do dia e hora em que as leituras foram realizadas, acreditou-se que dados como o Caudal Natural da barragem da Bouçã e o Caudal Afluente da barragem do Cabril, teriam algum impacto na Cota de Montante residente na Bouçã. Visto o Caudal Natural não ser uma leitura que se consiga obter através de sensores, teve de se recorrer a um cálculo da seguinte forma:

Calculo do Caudal Natural da Bouçã

$$\mathbf{caudNat}_{Bouca} = \mathbf{caudAflu}_{Bouca} - \mathbf{caudLanc}_{Cabril(n-1)} \quad (4.1)$$

sendo que:

$$\mathbf{caudLanc}_{Cabril} = \mathbf{caudTurb}_{Cabril} - \mathbf{caudDesc}_{Cabril} - \mathbf{caudEco}_{Cabril} \quad (4.2)$$

O Caudal Lançado do Cabril é calculado a partir da soma do Turbinado, do Descarregado e do Ecológico. Sabendo o valor do Lançado da hora anterior, tendo em consideração o tempo que leva à água para chegar à Bouçã, podemos saber o Caudal Natural da Bouçã subtraindo ao seu Afluente, o Lançado do Cabril.

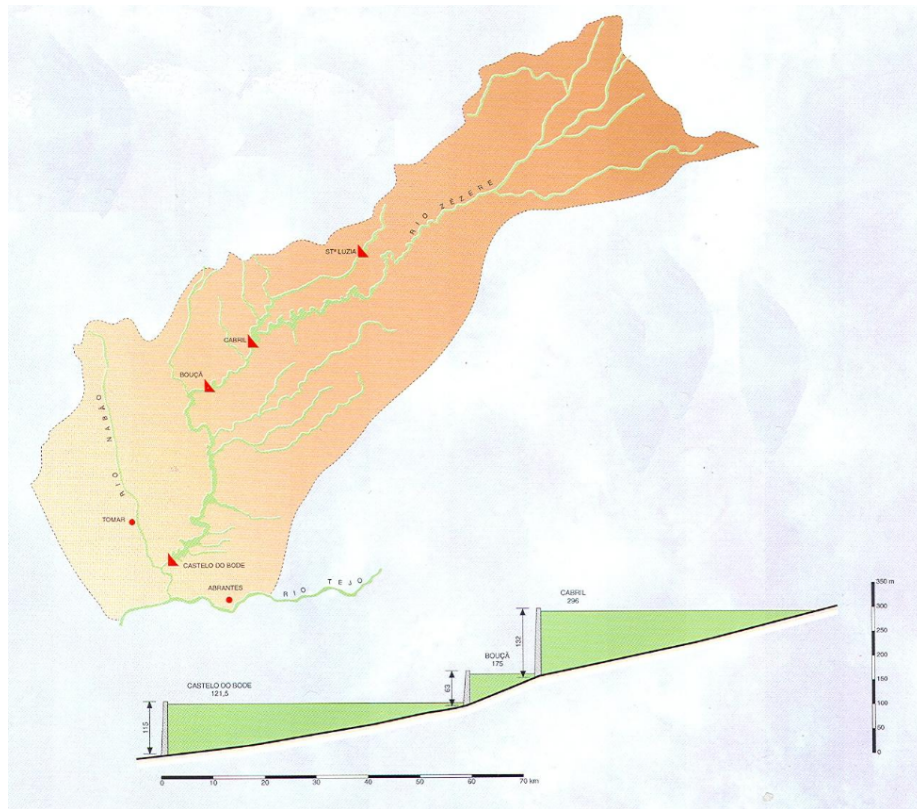


Figura 4.5: Sequência do Rio Zêzere [1]

Pelo facto de não existir Caudal Lançado para a barragem do Cabril, o seu Caudal Afluente é considerado o Natural. Sendo assim, esse valor pode ser obtido diretamente das leituras.

4.1.4 Correlação dos parâmetros

De modo a verificar como os parâmetros inseridos no modelo se relacionam entre si, realizou-se um conjunto de validações. Sendo as variáveis relativas aos caudais de entrada e saída aquelas com mais ênfase no sistema, inicialmente validou-se o comportamento da Cota de Montante em relação ao Caudal. Para tal foram analisadas 2 situações completamente distintas. No primeiro cenário é apresentado um caso que ocorreu na Bouça, entre o dia 20 de Dezembro de 2019 e o dia 23 de Dezembro de 2019, no qual se verificou uma enorme pluviosidade, levando à subida do nível da Cota de Montante. Consequentemente a necessidade de libertar água foi crucial para a segurança da infraestrutura.

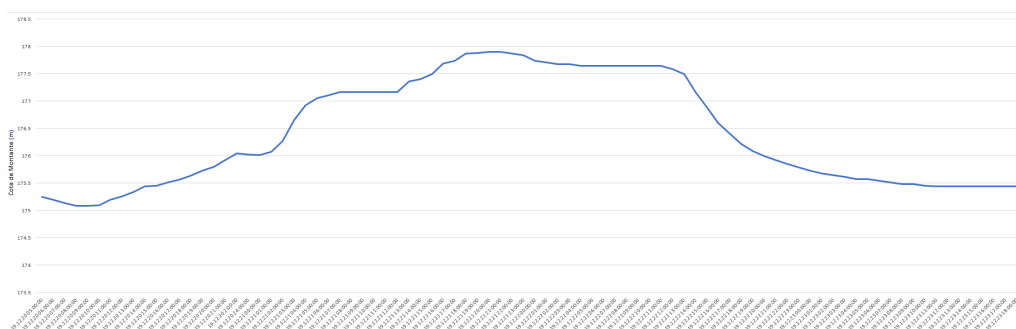


Figura 4.6: Cenário 1 da Variação horária do nível da Cota de Montante, em metros, da Barragem da Bouçã

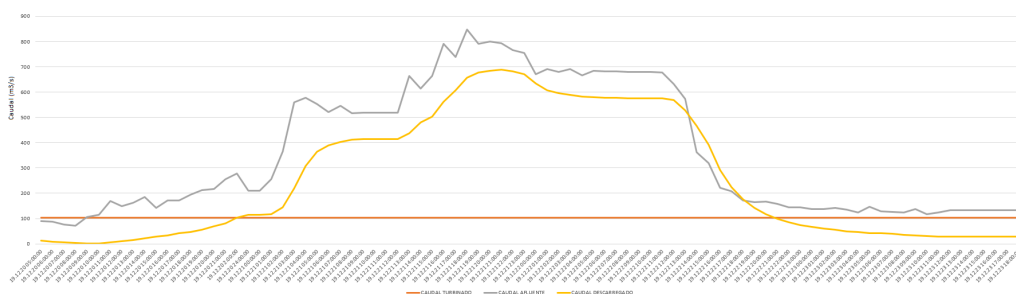


Figura 4.7: Cenário 1 da Variação horária dos Caudais, em metros cúbicos por segundo, da Barragem da Bouçã

Como é possível observar os níveis de Caudal Afluente, ou seja, o Caudal Natural mais o Caudal proveniente da barragem do Cabril, foram bastante significativos, levando a um aumento do volume de Caudal Descarregado. Simultaneamente, os níveis de Caudal Turbinado encontravam-se fixos na escala dos 100 m³/s. Caso este descarregamento não tivesse ocorrido, o nível da Cota de Montante máxima estabelecido, designado de Nível Pleno de Armazenamento, teria sido ultrapassado, provocando com que água começasse a transbordar pelo topo da barragem.

Já o segundo cenário, representa o que seria espetável de ocorrer numa situação mais habitual:

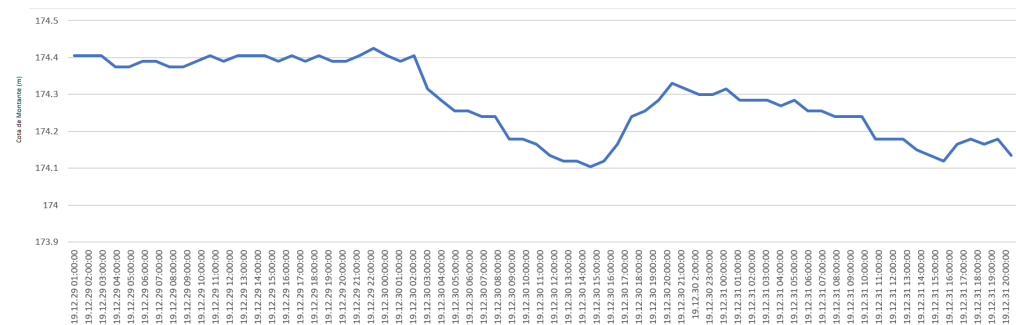


Figura 4.8: Cenário 2 da Variação horária do nível da Cota de Montante, em metros, da Barragem da Bouçã

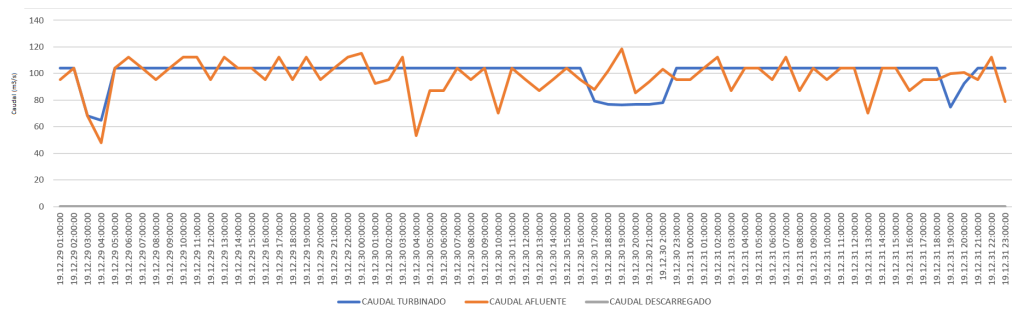


Figura 4.9: Cenário 2 da Variação horária dos Caudais, em metros cúbicos por segundo, da Barragem da Bouçã

Verificando-se que de facto não existiram descarregamentos na barragem, as variações apenas ocorreram nos valores de Caudal Turbinado e Caudal Afluente, o que por si não tem grande impacto no nível da Cota de Montante.

Com base neste tipo de conclusões, a rede neuronal ao ser treinada sobre este conjunto de parâmetros, consegue realizar múltiplas correlações, facilitando a estimação de resultados. O facto de ocorrer um aumento significativo na quantidade de Caudal Descarregado, será sempre um indicador chave para possíveis subidas repentinas do nível da cota. Da mesma forma, o Caudal Afluente revela a quantidade de água que se encontra a chegar à barragem, implicando uma subida da cota, sendo que para contrabalançar esse efeito é inserida a medida de Caudal Turbinado, significando que existe água a sair da barragem.

Ao realizar a identificação dos parâmetros existentes no sistema, constatou-se que a Cota de Jusante, ou seja, a altura da água existente a jusante da barragem, encontra-se directamente interligada com a de Montante, podendo servir de suporte para a estimação da variação das cotas. Em situações que a Cota de Jusante não varia, podemos assumir que a de Montante também não:

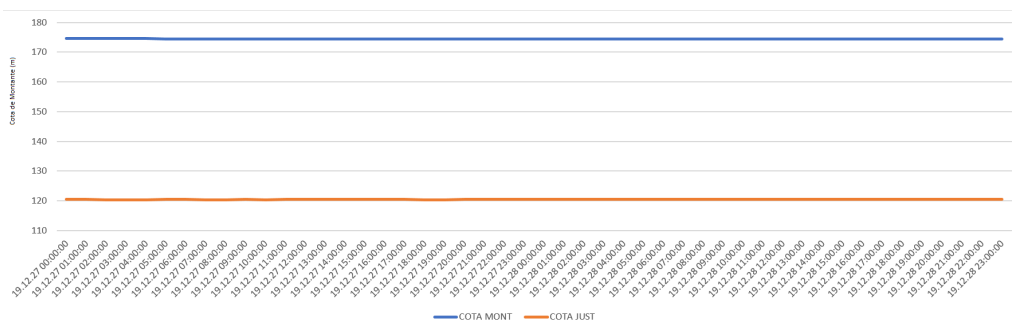


Figura 4.10: Cenário 1 da Variação horária do nível da Cota de Jusante, em metros, da Barragem da Bouçã

Já em cenários que se verificam variações relevantes da Cota de Jusante, muito provavelmente isso implica que a de Montante sentiu o mesmo efeito:

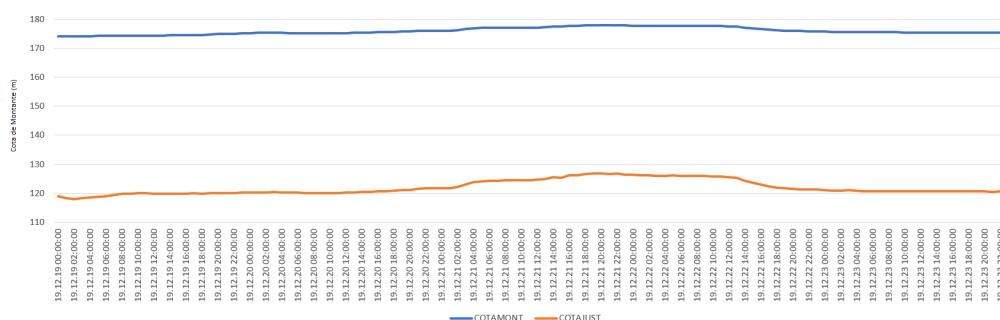


Figura 4.11: Cenário 2 da Variação horária do nível da Cota de Jusante, em metros, da Barragem da Bouçã

Verificando os resultados desta análise, decidiu-se assim integrar a Cota de Jusante no *dataset* de treino do modelo de estimação de estados.

4.2 Conclusões

Através da análise de dados realizada, concluiu-se que existem bastantes especificidades a ter em consideração ao estudar as características de uma infraestrutura hidroelétrica. Com base nos resultados obtidos, decidiu-se seguir com a criação de um *dataset* de treino constituído pelos seguintes parâmetros:

- Dia
- Hora
- Dados da Barragem da Bouçã:
 - Cota de Montante
 - Cota de Jusante
 - Caudal Turbinado
 - Caudal Afluente
 - Caudal Descarregado
 - Caudal Natural
- Dados da Barragem do Cabril:
 - Caudal Afluente

O modelo será então treinado com os eventos horários, registados no GPinfo, que ocorreram entre o dia 1 de Janeiro de 2000 e o dia 1 de Janeiro de 2020, reunindo assim um total de 20 anos de dados.

Capítulo 5

Desenvolvimento e Integração

Nesta secção será apresentado o desenvolvimento do modelo de estimação de Cotas de Montante, juntamente com as adversidades encontradas ao longo do tempo. O objetivo primordial será sempre a validação da atual leitura da medida da Cota de Montante. Realizando esta estimação com sucesso, haverá uma readaptação do modelo, para prever a evolução dessa mesma cota, para um dado período de tempo. Todo este processo deverá encontrar-se completamente automatizado, garantindo que sempre que existe uma nova medida, esta é instantaneamente validada, diminuindo assim o tempo de reação por parte das equipas operacionais.

5.1 Ambiente de Desenvolvimento

A linguagem de programação *Python* foi a selecionada para o desenvolvimento desta solução. Por esse motivo optou-se pela utilização do WinPython [31], que consiste numa distribuição portátil e *open-source* da linguagem *Python*, projetado para aplicações de *data-science*, pelo facto de conter diversas bibliotecas que ajudam na criação de projetos de Machine Learning e Inteligência Artificial. O WinPython proporciona um conjunto de IDE's para serem utilizadas, sendo o Spyder [13] o escolhido para este projeto.

O Spyder caracteriza-se como um poderoso ambiente de desenvolvimento integrado, especificamente projetado para cientistas, engenheiros e *data analysts*, sendo uma ferramenta direcionada tanto para desenvolvimento, como para a análise e exploração de dados.

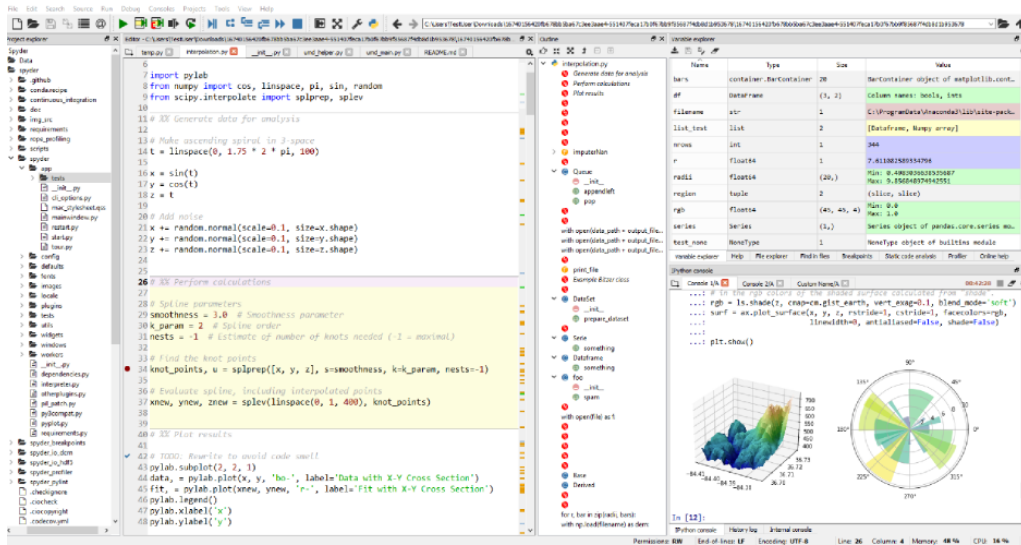


Figura 5.1: Demonstração do IDE Spyder [13]

5.2 Data

Como referido no Capítulo 3.5, os dados armazenados na base de dados GPinfo serão utilizados como *input* para o modelo, de modo a que este consiga treinar a sua rede neuronal. Esta base de dados encontra-se montada no Oracle SQL Developer, sendo que para construir o *dataset* desejado, certos filtros tiveram de ser aplicados.

Para a recolha dos dados necessários, relativos à barragens da Bouçã, a seguinte *query* foi aplicada:

```

Query Bouçã
SELECT
    DATA,
    COTAMONT,
    COTAJUS,
    CAUDTURB,
    CAUDAFLUT,
    CAUDESC,
    HORA
FROM
    CENTRAIS_HORARIOS
WHERE
    SIGLA like 'BC'
    AND DATA >= '00.01.01'
ORDER BY
    DATA, HORA

```

Algoritmo 1: Query de recolha de dados da Barragem da Bouçã

Retornando assim a Data e a Hora, a Cota de Montante e de Jusante, e por fim, o Caudal Turbinado, Afluente e Descarregado. Todos estes *items* foram filtrados de forma

apenas aparecerem aqueles relativos à Bouçã, cuja data seja superior a 1 de Janeiro de 2000.

Para os dados relativos à barragem do Cabril, a seguinte *query* foi utilizada:

Query Cabril

```
SELECT
  DATA,
  CAUDTURB,
  CAUDESC,
  CAUDECOLOG,
  HORA
FROM
  CENTRAIS_HORARIOS
WHERE
  SIGLA like 'CR'
  AND DATA >= '00.01.01'
ORDER BY
  DATA, HORA
```

Algoritmo 2: Query de recolha de dados da Barragem do Cabril

Esta *query* apenas retorna a Data e a Hora, juntamente com os valores do Caudal Turbinado, Descarregado e Ecológico, de forma a poder-se calcular o Caudal Lançado da barragem do Cabril. De relembrar, como referido no Capítulo 4.1.3, que este caudal servirá para descobrir o Caudal Natural da Bouçã, subtraindo ao seu Afluente, o Caudal Lançado da barragem do Cabril na hora anterior.

Obtidos todos os dados necessários, segue-se para criação do *dataset*, no formato *.csv*, para que este seja interpretado pelo algoritmo de estimação de estados. O *template* do ficheiro é o seguinte:

Tabela 5.1: Configuração base do *dataset* referente à Barragem da Bouçã

Data	Cota Montante	Cota Jusante	Caud. Turbinado	Caud. Afluente	Caud. Descarregado	Caud. Afluente CR	Caud. Natural	Hora
20.01.01	175.35	120.33	0	24.48	18.68	327.11	43.76	3
20.01.01	175.40	120.35	0	25.01	18.90	327.00	40.76	4
20.01.01	175.60	120.38	0	24.99	18.93	327.23	41.76	5
20.01.01	175.50	120.34	0	24.50	18.90	327.10	41.6	6
...
...
...

5.3 Desenvolvimento

Com a preparação do *dataset* concluída, segue-se o desenvolvimento de um algoritmo dividido nas seguintes secções:

1. Importação das bibliotecas;
2. Leitura do *dataset*;
3. Reformulação dos dados;

4. Repartição do *dataset*;
5. Criação e treino da rede neuronal;
6. Geração de previsões;
7. Apresentação dos resultados.

5.3.1 Inicializações

Prévio ao início da fase de desenvolvimento, um conjunto de bibliotecas, necessárias para a criação do modelo, tiveram de ser importadas.

Para o tratamento e visualização de dados as seguintes bibliotecas foram utilizadas:

Import's

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import rc
import pandas as pd
from pandas.plotting import register_matplotlib_converters
import seaborn as sns
from pylab import rcParams
```

Algoritmo 3: Primeiro conjunto de *import's*

A biblioteca NumPy vai ser fundamental para converter os dados do *dataset* em matrizes multidimensionais, o Matplotlib e o Seaborn possibilitam a visualização dos dados e a criação de gráficos, e o Pandas é adequado para a manipulação e análise de dados, tabelas numéricas e séries temporais.

No que toca à criação da rede neuronal, foi importado o tensorflow, juntamente com outras bibliotecas adicionais:

Import's

```
import tensorflow as tf
from tensorflow import keras
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import RobustScaler
```

Algoritmo 4: Segundo conjunto de *import's*

O Keras consiste na biblioteca de redes neuronais que vai ser utilizada neste projeto, e o Sklearn contém múltiplas ferramentas de *Machine Learning*, possibilitando classificação, regressão, *clustering* entre outras tarefas.

5.3.2 Repartição do conjunto de dados

Com todas as bibliotecas importadas, segue-se à interpretação e divisão do *dataset* num conjunto de treino e outro conjunto de testes. Além disso, indexou-se a coluna do *timestamp*, possibilitando a criação de três colunas adicionais. Gerou-se uma coluna com

dia do ano, outra com o dia do mês, e finalmente, com o respetivo mês em que a leitura foi realizada.

Leitura e distribuição do *dataset*

```
df = pd.read_csv('bc_dataset.csv')

df.DATA = pd.to_datetime(df.DATA)
df = df.set_index('DATA')

df['Dayyear'] = df.index.dayofyear
df['Daymonth'] = df.index.day
df['Month'] = df.index.month

train_size = int(len(df)*0.95)
test_size = len(df) - train_size

train, test = df.iloc[0:train_size], df.iloc[train_size:len(df)]
```

Algoritmo 5: Leitura e distribuição dos dados

De modo a tornar este modelo o mais preciso possível, o seu *dataset* de treino conteve 95% dos dados, enquanto que os restantes 5% permaneceram para validações e realização de testes. Por motivos de *performance* e tempo de processamento, apesar de terem sido extraídos dados a datar de Janeiro de 2000, apenas vão ser utilizados os eventos que ocorreram a partir de Janeiro de 2010. Ao somente utilizar 10 anos de registos, consegue-se diminuir o tempo de treino em metade, mantendo acontecimentos suficientes para garantir a eficácia do modelo. O treino decorreu utilizando um computador equipado com um processador Intel Core i5-6500, com 16GB de RAM e uma placa gráfica GTX 1060 de 6Gb.

5.3.3 Implementação da rede neuronal

O *dataset* aplicado contém uma ampla variedade de valores. De modo a garantir eficiência no treino do modelo, foi necessário escalar os dados antes destes serem inseridos na rede neuronal.

Para isso criou-se um *object scaler* para os sinais de entrada e outro para o de saída:

Transformação da escala dos sinais de entrada e de saída *dataset*

```

sinaisEntrada =
    ['COTAJUS', 'CAUDTURB', 'CAUDAFLUEN', 'CAUDESC']

scalerEntrada = RobustScaler()
scalerSaida = RobustScaler()

scalerEntrada = scalerEntrada.fit(train[sinaisEntrada].to_numpy())
scalerSaida = scalerSaida.fit(train[['COTAMONT']])

train.loc[:,sinaisEntrada] =
    scalerEntrada.transform(train[sinaisEntrada].to_numpy())
train['COTAMONT'] = scalerSaida.transform(train[['COTAMONT']])

test.loc[:,sinaisEntrada] =
    scalerEntrada.transform(test[sinaisEntrada].to_numpy())
test['COTAMONT'] = scalerSaida.transform(test[['COTAMONT']])

```

Algoritmo 6: Transformação do *dataset*

A conversão efetuada com o *RobustScaler* resultou com que o *dataset* sofre-se as seguintes alterações:

Tabela 5.2: Nova escala do *dataset* da Barragem da Bouçã

Data	Cota Montante	Cota Jusante	Caud. Turbinado	Caud. Afluente	Caud. Descarregado
10.01.01	4.326923077	0.351585014409221	-0.077526534	0.11519407933182	18.68
10.01.01	4.673076923	0.740634006	0.446585141	1.204507207	20.46
10.01.01	4.673076923	0.752161383	0.485348408	0.707959843	26.7
...
...
20.01.01	6.32692307692333	1.132564841	0.522381172	1.395898152	88.61
20.01.01	6.403846154	1.132564841	0.522381172	1.581013985	86.78
20.01.01	6.326923077	1.132564841	0.522381172	1.290267084	85.44
...
...

Como nos encontramos a trabalhar com um conjunto de dados que contém milhares de linhas, de modo a que seja possível treinar o modelo com toda essa informação, as séries temporais existentes vão ser divididas em sequências menores. O período entre cada sequência vai representar a janela temporal de dados no qual a rede neuronal vai treinar.

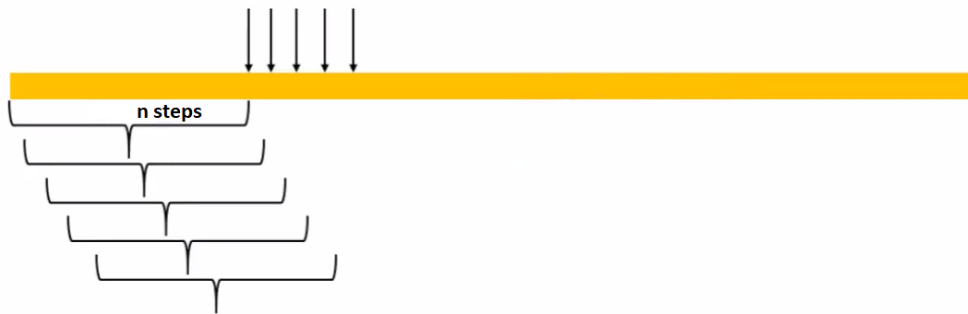


Figura 5.2: Representação da divisão das series temporárias na rede neuronal.

Os dados vão ser lidos em porções de n steps, predefinidos no algoritmo, ou seja, a rede neuronal analisa e treina durante o período n e realiza a previsão para a entrada $n+1$. Na seguinte análise a primeira entrada é ignorada, passando a receber como *input* os restantes dados, até a linha $n+1$, sendo este ciclo repetido até que todos os dados tenham sido observados. Esta metodologia pode ser designada por "*Sequence Classification*", pois a partir de uma sequência de entrada, vamos estimar o futuro resultado. Quanto maior o valor do período da análise, maior será o processamento por parte da GPU e menor será a certeza no valor estimado pelo facto de existirem mais parâmetros a ter em consideração para se chegar a um resultado final.

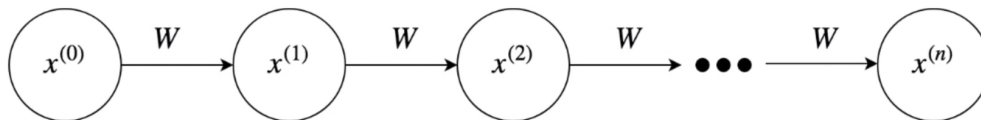


Figura 5.3: Exemplo do processamento das sequências.

Com a imagem ilustrada compreende-se que o tamanho das sequências teoricamente pode ser "infinito", isto é, depois de n ocorrências, o valor de x será $x(n)$, e acrescentado-lhe a sua escala de crescimento W , para a situação em que o número de *steps* é elevado, o desvio do resultado final relativamente, ao valor real, poderá encontrar-se bastante desfasado.

Além disso, existem cenários em que apenas precisamos de olhar para informações atuais para prever o *output* desejado. Para esses casos, onde a diferença entre as informações relevantes e a sequência observada, é pequena, as Recurrent Neural Network's podem aprender a usar eventos passados para chegar ao seu resultado de forma mais eficiente. É também possível que o intervalo de tempo entre a informação relevante da sequência de dados e o ponto para o qual ela é de facto necessária, seja algo dispersa, o que leva a que a rede seja incapaz de aprender a interligar os eventos [14].

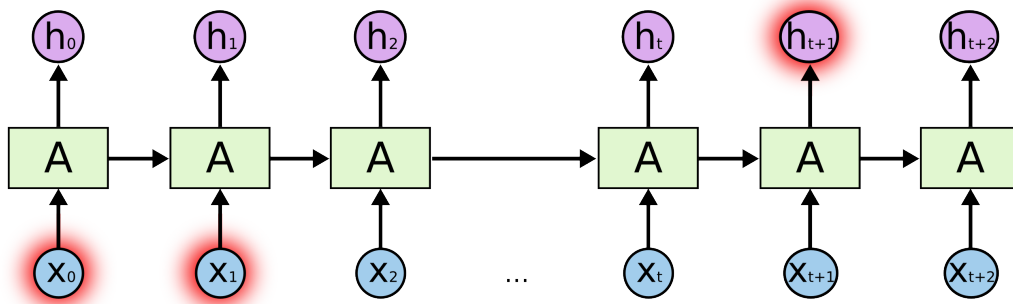


Figura 5.4: Dependência a longo prazos nas Recurrent Neural Network's [14]

Para esses casos foram desenvolvidas as LSTM Networks, projetadas explicitamente para evitar problemas de dependências a longo prazo, sendo este praticamente o seu modo de funcionamento padrão.

Por todos estes motivos vai-se treinar o modelo com diferentes *timesteps*, verificando qual deles apresenta as melhores previsões.

5.3.3.1 Criação da rede neuronal para o modelo de estimação de estados

O modelo de estimação de cotas foi criado utilizando uma das arquiteturas das *Recurrent Neural Network's* aplicada no campo do *Deep Learning*, designada por *Long Short-Term Memory*, ou como previamente referido, LSTM.

Ao contrario de outros modelos, a arquitetura LSTM pode aprender a reconhecer padrões sensíveis ao contexto em que este foi implementado, sendo esse um dos grandes factores da sua escolha. A variação da Cota de Montante nas barragens tende a ter comportamentos bastante semelhantes à medida que o tempo vai passando, sendo que regra geral os impulsos que levam à sua subida, ou descida, tendem a ser os mesmos.

As redes neurais recorrentes têm a forma de uma sequência de módulos repetidos, possuindo uma estrutura simples, como uma única camada *tanh*.

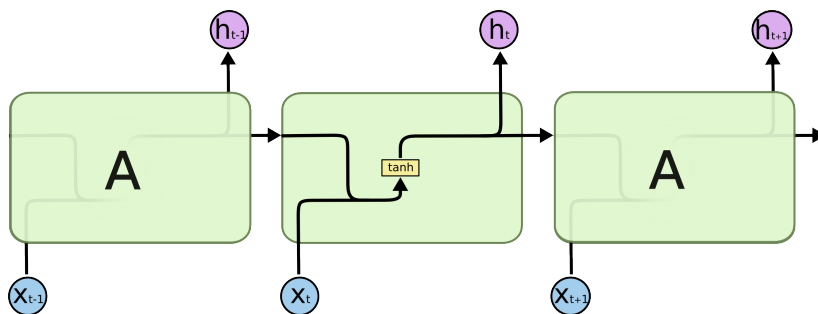


Figura 5.5: Módulo de repetição de uma RNN padrão. [14]

As camadas *tanh* são como uma sigmóide logística, diferindo no intervalo da sua função, que vai de -1 a 1, apresentado a vantagem no facto que as entradas negativas

serão mapeadas como 'fortemente negativas' e as entradas zero mapeadas perto de zero no gráfico [15]:

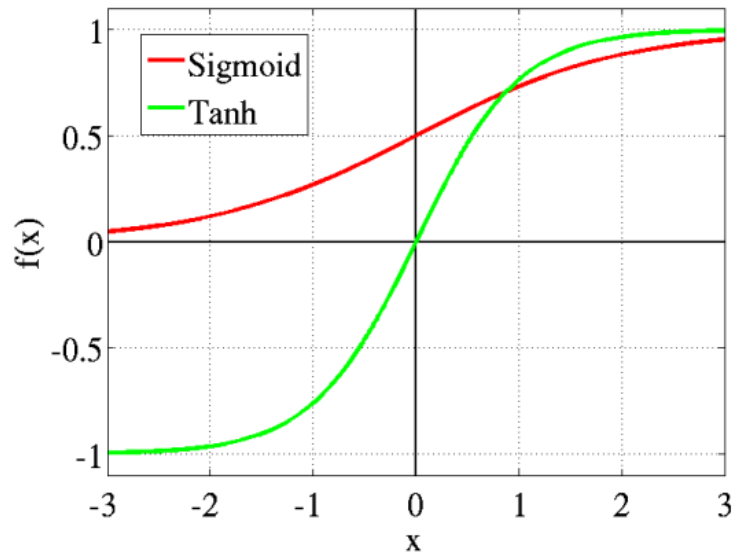


Figura 5.6: Comparação de uma Sigmóide em relação a uma camada *tanh* [15]

As redes LSTM também possuem uma estrutura em cadeia, só que em vez de uma única camada, existem quatro, que interagem entre si.

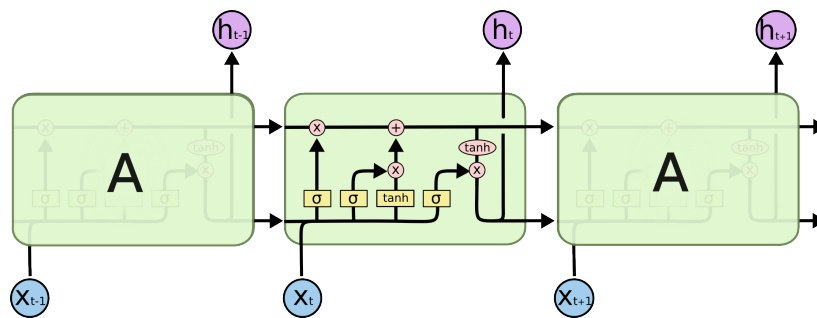


Figura 5.7: O módulo de repetição presente numa rede LSTM [14]

Como referido no Capítulo 3.2.5, este tipo de redes contém uma célula de estado, que percorre toda a cadeia com apenas algumas menores interações lineares. A sigmóide gera números entre 0 e 1, representando quanto de cada componente deve ser liberado, sendo que o valor 0 significa 'não deixar passar nada' e o valor 1 'deixar passar tudo' [14].

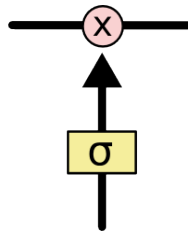


Figura 5.8: Camada sigmóide [14]

As redes LSTM apresentam todas estas camadas de forma a poder proteger e controlar a sua célula estado, garantido assim que apenas a informação fulcral é transmitida para a saída.

Para que se pudesse dar início à criação do modelo, o número de *timesteps* foi então inicializado e o *dataset* distribuído em dois *arrays* de entrada, *X_train* e *X_test*, e dois *arrays* de saída, *y_train* e *y_test*.

Modelo de estimação

```

model = keras.Sequential()

model.add(
    keras.layers.Bidirectional(
        keras.layers.LSTM(
            units=128,
            input_shape=(X_train.shape[1], X_train.shape[2])
        )
    )
)

model.add(keras.layers.Dropout(rate=0.2))
model.add(keras.layers.Dense(units=1))
model.compile(loss='mean_squared_error', optimizer='adam')

```

Algoritmo 7: Criação do modelo de estimação de estados

Um modelo sequencial é inicializado a partir da biblioteca Keras, contendo uma primeira camada bidirecional, o que implica que para além de observar o histórico do conjunto de dados, também verifica, e aprende, o que as suas variações causam no futuro. Dentro da camada bidirecional é chamada a camada com a rede LSTM, configurada com 128 neurónios e a forma da matriz que vai receber de *input*.

Fora do modelo sequencial, é chamada uma camada de *Dropout* que consiste num método de regularização em que as conexões de entrada e recorrentes são probabilisticamente excluídas da ativação e das atualizações de peso, ao longo do treino da rede, o que leva a uma redução de ajustes em excesso, melhorando assim o desempenho do modelo.

Para terminar, é adicionada uma *Dense layer*, que consiste numa camada totalmente conetada que segue as camadas da rede LSTM e é usada para gerar a previsão. Visto apenas estar a ser previsto um único valor numérico, apenas irá conter um único neurónio.

No final o modelo é compilado com o "Mean Squared Error" como a função de perda e como otimizador, o "Adam". A função de perda MSE é bastante utilizada na realização de regressões pelo facto de ser sensível a valores discrepantes, gerando a perda através da média das diferenças quadráticas, entre os valores reais e os previstos, de todos os dados supervisionados [32]. O otimizador *Adam* é um algoritmo adaptativo de otimização da taxa de aprendizagem do modelo. Ele foi desenvolvido especificamente para o treino de redes neuronais, aproveitando o método de taxas de aprendizagem adaptável, para encontrar taxas de aprendizagem individuais para cada parâmetro [33].

Aplicadas todas estas configurações, o modelo foi treinado múltiplas vezes com um número diferente de *epochs*, de forma a averiguar qual deles retornava o melhor conjunto de previsões. O número de *epochs* vai definir o número de cortes aplicados por todo o conjunto de dados, posteriormente usados para separar o treino em fases distintas.

Secção do algoritmo de execução do treino do modelo de estimação de estados

```

model.fit(
    X_train, y_train,
    epochs= XXX ,
    batch_size=32,
    validation_split=0.1,
    shuffle=False
)
model.save('EstimadorEstados.h5')

```

Algoritmo 8: Treino do modelo de estimação de estados

5.3.4 Fluxograma dos algoritmos

Este projeto resultou na criação de duas *scripts* distintas. Uma para a criação e treino do modelo e outra para o seu carregamento e execução das previsões.

Foi necessário garantir que após a fase de treino, o modelo seria instantaneamente guardado de modo a que esta não tenha que ser repetida. O treino de um modelo de estimação pode ser bastante demorado, daí não ser eficiente a sua realização sempre que tencionemos gerar novas previsões.

O modo como o algoritmo de criação e treino se encontra construído é o seguinte:

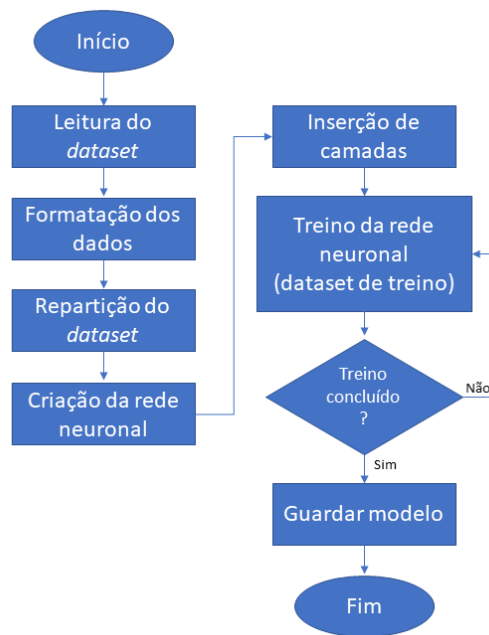


Figura 5.9: Fluxograma da criação do modelo de estimação de estados.

O algoritmo de carregamento e execução é bastante semelhante ao de criação, diferenciando-se pelo facto deste carregar o modelo e gerar as previsões para o conjunto de dados que recebeu à entrada.

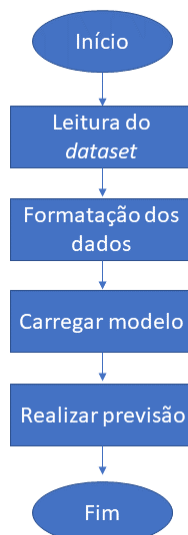


Figura 5.10: Fluxograma da execução do modelo de estimação de estados.

5.3.5 Previsão de cotas

A previsão de cotas foi uma característica adicional, que acrescenta valor à operação, daí o seu desenvolvimento. Para isso aplicou-se uma previsão de séries temporais, mais uma vez, usando redes LSTM, idênticas às do modelo de estimação, que recebe a última data de *input* do *dataset* e acrescenta-lhe um período de horas adicional [34]:

Secção do algoritmo de inclusão de um período futuro no *dataset*

```
from pandas.tseries.offsets import DateOffset

add_dates = [df.index[-1] + DateOffset(hours=x) for x in range(0, XX ) ]
future_dates = pd.DataFrame(index=add_dates[1:],columns=df.columns)
```

Algoritmo 9: Inclusão de um período futuro no *dataset*

O parâmetro *XX*, representa o número de horas no futuro que queremos prever.

Posterior à adição destes novos registos, é criada uma lista onde alocar os valores simulados, e estes adicionados a uma nova coluna no *dataset*, denominada de "*Prediction*":

Secção do algoritmo de previsão futura do conjunto de horas imputadas.

```
df_predict = pd.DataFrame(scaler.inverse_transform(pred_list),
                          index=future_dates[-n_input:].index, columns=['Prediction'])

df_proj = pd.concat([df,df_predict], axis=1)
```

Algoritmo 10: Prevendo para além do conjunto de dados

O período de horas a ser simulado foi sendo ajustado de modo a verificar qual o conjunto que apresentava valores mais fidedignos, sendo estes resultados demonstrados no Capítulo 6.

Capítulo 6

Resultados e Validações

O principal objetivo deste capítulo consiste na verificação do feedback por parte do modelo de estimação de estados relativamente às medidas selecionadas para treinar a rede neuronal. Nas próximas secções serão apresentados os resultados obtidos através de diferentes testes de validação e calibração do algoritmo.

6.1 Testes de validação

Ao longo de todo o desenvolvimento deste projeto, foram realizados vários testes sobre a rede neuronal, de forma a avaliar como esta se comportaria relativamente ao seu conjunto de *inputs*, e conseqüentemente, quais os resultados apresentados, em função das inicializações aplicadas. De modo a obter um *output* confiável, múltiplos testes de execução foram realizados, permitindo o ajuste do estimador. Para além dos ensaios aplicados ao modelo de estimação, também serão evidenciados os testes realizados sobre o modelo de previsões futuras.

Primordialmente, serão apresentados os resultados obtidos com uma rede neuronal treinada com todos os parâmetros extraídos. As seguintes figuras encontram-se apresentadas com os valores escalados, aplicados pelo *RobustScaler*, simétricos aos reais, sendo que o passo para a reconversão da escala apenas foi realizado posteriormente à consolidação do modelo.

- Rede Neuronal treinada com 10 *epochs* e 120 *steps* entre amostragem:

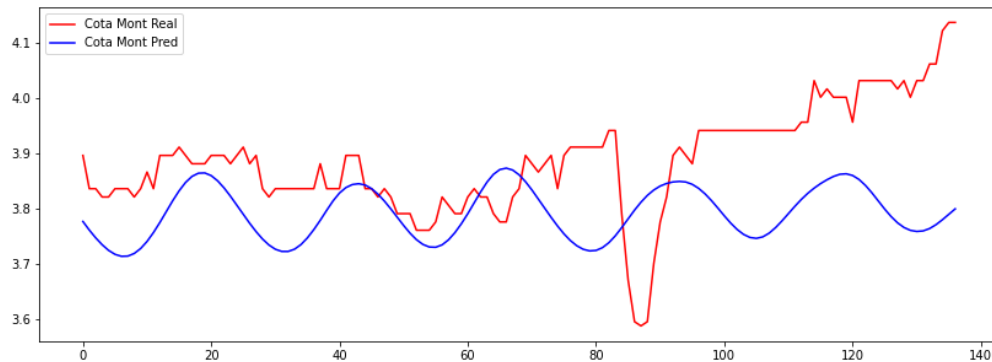


Figura 6.1: Previsões realizadas com 10 *epochs* e 120 *steps*

De modo a verificar o comportamento do modelo, o primeiro conjunto de simulações foram realizadas com poucos *epochs* e um período de amostragem significativamente elevado. As simulações obtidas demonstraram-se completamente desfasadas da realidade. Com base nestes resultados prosseguiu-se à diminuição do período entre cada amostra de treino e o aumento do número de *epochs*.

- Rede Neuronal treinada com 100 *epochs* e 60 *steps* entre amostragem:

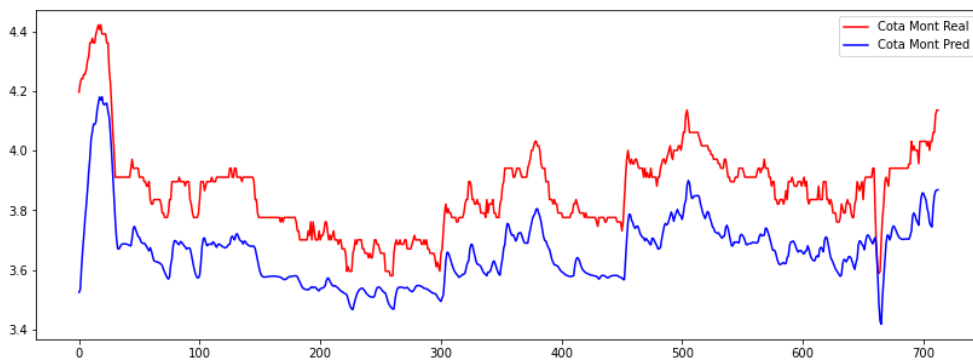


Figura 6.2: Previsões realizadas com 100 *epochs* e 60 *steps*

A diminuição do número de *steps* e aumento de *epochs* resultou numa previsão bastante semelhante à realidade. Ocorrendo um treino mais prolongado, logicamente menor será o *loss*, isto é, menor será o valor que indica o quão ineficiente foi

a previsão do modelo, evidenciando assim que apenas com um elevado número de *epochs* conseguimos aproximar a previsão à realidade.

Para verificar o impacto do período de amostragem nos resultados do modelo, prosseguiu-se para o treino de um modelo, com uma redução drástica do número de *steps* recolhidos.

- Rede Neuronal treinada com 100 *epochs* e 6 *steps* entre amostragem:

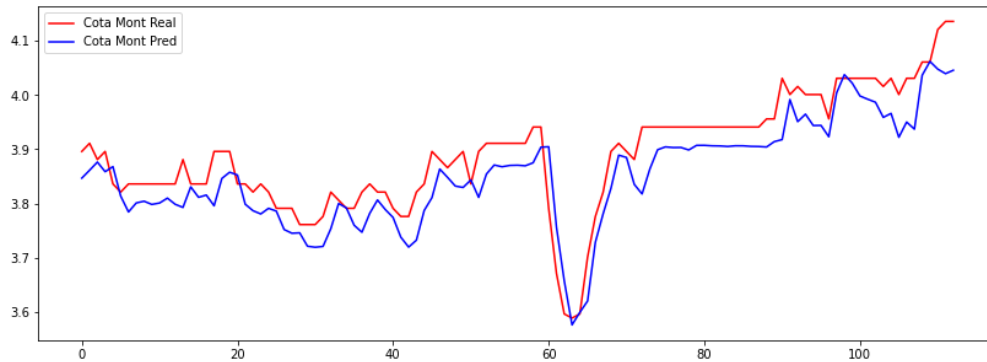


Figura 6.3: Previsões realizadas com 100 *epochs* e 6 *steps*

Os resultados desta simulação foram bastante positivos, existindo assim um menor desvio entre os dados previstos e os reais, levando à assunção que para uma previsão eficaz do valor da Cota de Montante, não pode ser reunida uma amostra de dados demasiado grande.

Compreendendo-se nesta primeira fase de testes, a relação entre o número de *epochs* e o período de amostragem predefinido, prosseguiu-se à verificação do comportamento do modelo relativamente à remoção do Caudal Natural da barragem da Bouça como um variável de entrada. Decidiu-se remover esta medida pelo facto de não ter uma influência direta sobre o nível da Cota de Montante.

- Rede Neuronal treinada com 200 *epochs* e 24 *steps* entre amostragem:

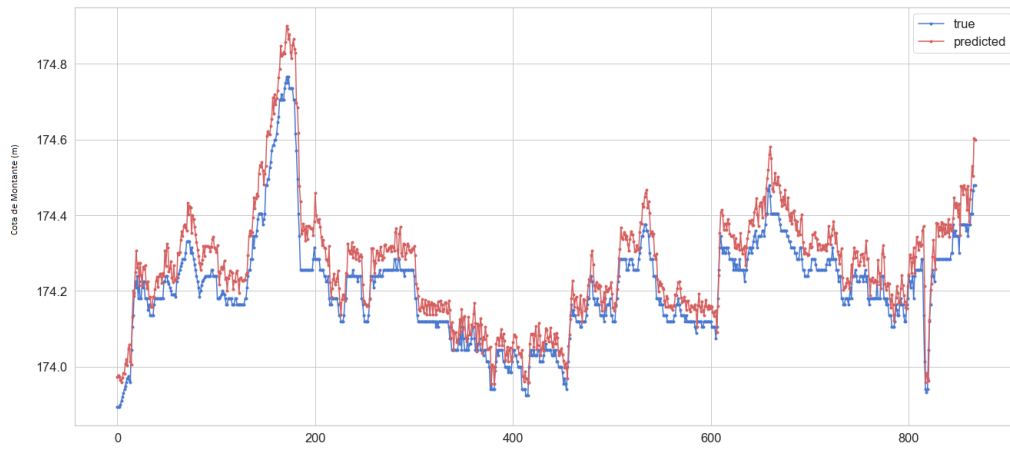


Figura 6.4: Previsão horária do nível da Cota de Montante, em metros, realizada com 200 *epochs* e 24 *steps*

- Rede Neuronal treinada com 500 *epochs* e 6 *steps* entre amostragem:

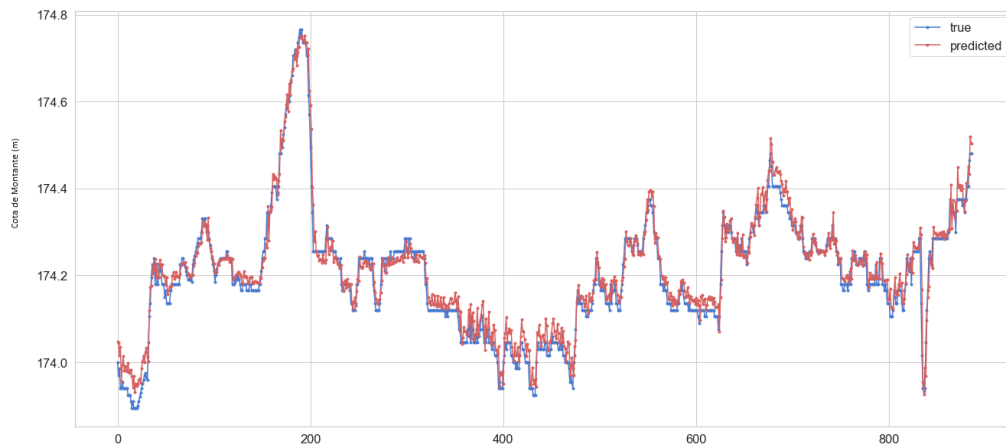


Figura 6.5: Previsão horária do nível da Cota de Montante, em metros, realizada com 500 *epochs* e 6 *steps*

O treino realizado com 500 *epochs* e 6 *steps* entre cada amostragem comprovou-se bastante compacto, existindo apenas ligeiros desvios em determinadas ocasiões:

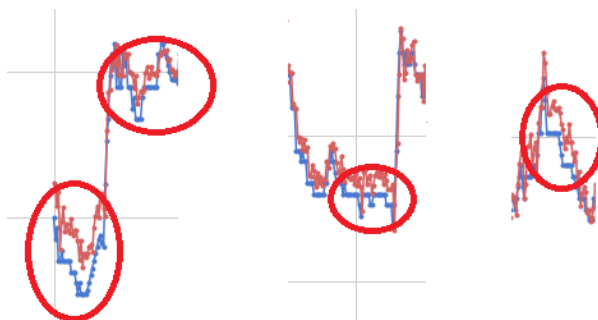


Figura 6.6: Desvios existentes na previsão com 500 *epochs* e 6 *steps*

Aplicando uma *script* de validação, com um respetivo *threshold* sobre o *output* do estimador, podemos verificar se estas diferenças justificam a emissão de um alerta para o CTCH.

Para evitar estes desvios e de modo a consolidar ainda mais o modelo, procedeu-se ao treino da rede com um período de amostragem de apenas 3 *steps*. Visto cada leitura presente no *dataset* encontrar-se com um desfasamento de 1 hora, estes 3 *steps* equivalem a que o modelo se baseie nas últimas 3 horas para chegar ao seu resultado.

Este último teste resultou assim no modelo final de estimação de estados:

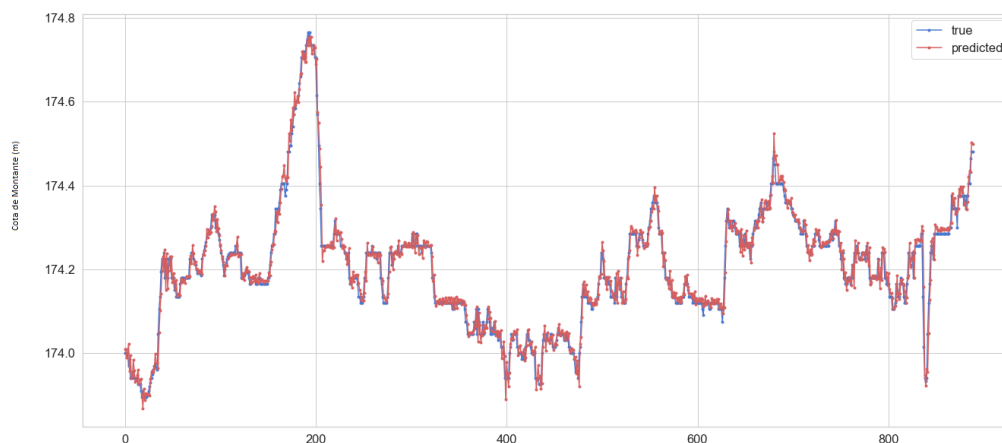


Figura 6.7: Previsão horária do nível da Cota de Montante, em metros, realizada com 500 *epochs* e 3 *steps*

Como é possível verificar, as previsões realizadas pelo modelo foram quase idênticas ao que ocorreu na realidade. Este resultado leva-nos a concluir que com base no histórico de eventos fornecido ao longo da fase de treino, a rede neuronal aprendeu corretamente os padrões por detrás do nível da Cota de Montante, possibilitando assim uma correlação eficaz das variáveis de entrada, resultando numa previsão com um desvio, para com o real, significativamente pequeno.

6.2 Desempenho do modelo

As seguintes amostras demonstram a consistência do modelo relativamente ao *dataset* de teste utilizado.

Apesar de se provar que a rede neuronal prevê extremamente bem o valor da Cota de Montante, existem sempre cenários anormais em que este pode vir a fracassar. A seguinte figura evidencia a diferença registada entre o valor da Cota de Montante real e o previsto, para um *timestamp* com um acontecimento invulgar:

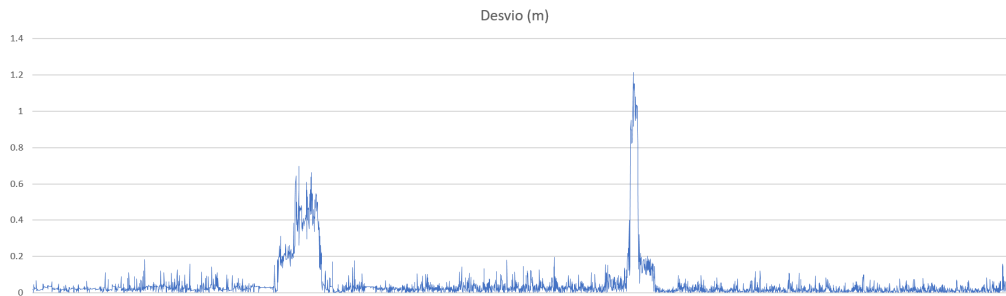


Figura 6.8: Desvio, em metros, entre a Cota de Montante Real e a Prevista

O desvio médio entre ambas as medidas é de 0.0484 metros, no entanto, como é possível de se verificar, existe um pico para o qual a diferença atinge 1.2 metros. Esse acontecimento teve início a 21 de Dezembro de 2019 e terminou a 22 de Dezembro de 2019, e foi este o respetivo acompanhamento do estimador de estados:

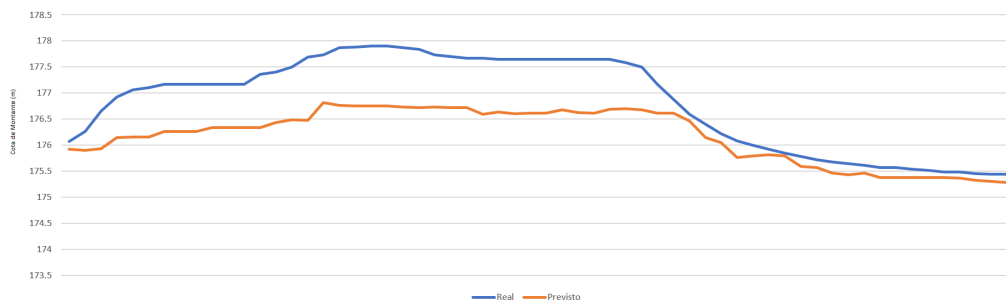


Figura 6.9: Variação, em metros, da Cota de Montante da Bouçã entre os dias 21/12/2019 e 22/12/2019

Regra geral, o nível da Cota de Montante na barragem da Bouçã encontra-se entre os 174 e os 175 metros, sendo que antes deste acontecimento, a última vez que a cota atingiu os 177 metros foi a 10 de Fevereiro de 2014 e os 178 metros a 24 de Novembro de 2006.

A curva de perda do modelo, ao longo do treino, foi a seguinte:

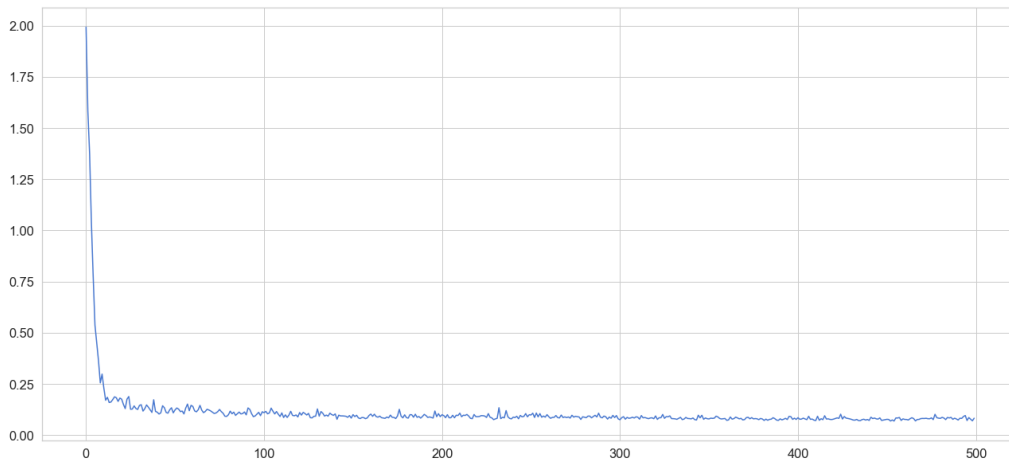


Figura 6.10: Curva de perda do modelo de Estimação de Estados

Sendo que a perda, ou *loss*, traduz-se pelo número que indica o quão inadequada foi a previsão, pode-se verificar que após 500 *epochs* esse valor é aproximadamente 0. A previsão do modelo é considerada de perfeita no momento em que este valor é absolutamente 0.

A função de perda é utilizada para otimizar algoritmos de *Machine Learning*, e a sua perda é calculada ao longo da fase de treino e validação. A sua interpretação é baseada no desempenho do modelo com base nesses dois conjuntos, ou seja, a soma dos erros cometidos nos conjuntos de treino e de validação.

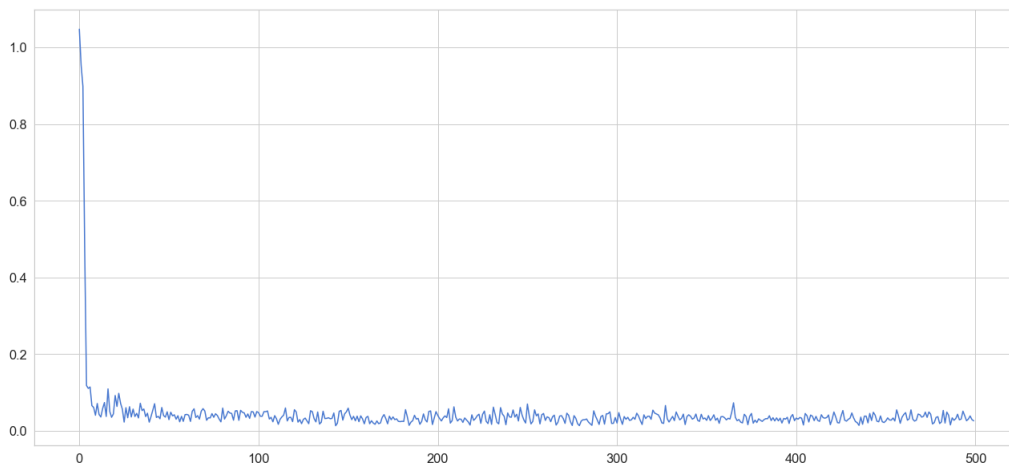


Figura 6.11: Curva de validação da perda do modelo de Estimação de Estados

6.3 Previsões futuras para além do dataset

Apesar de se demonstrar extremamente complicado prever com relativa exatidão a variação do nível de água armazenado na barragem, decidiu-se seguir em frente com esta implementação. Mesmo afastando-se do foco inicial do modelo de estimação de estados, a possibilidade de prever a variação da Cota de Montante um conjunto de horas além da atual, é um complemento bastante útil para a gestão do tempo de utilização dos grupos geradores de energia. Utilizando o ultimo conjunto de acontecimentos e comparando-os a prévios eventos semelhantes, pode ser possível prever o nível da Cota de Montante para um dado conjunto de horas.

Sendo que a ultima amostra utilizada no *dataset* continha o *timestamp* referente ao dia **02-03-2020 16:00:00**, a previsão do seguinte conjunto de horas resultou nas seguintes medidas:

Tabela 6.1: Resultados das previsões futuras além do *dataset* (1)

Data	Cota Prevista	Cota Real
02-03-2020 17:00:00	174.48	174.48
02-03-2020 18:00:00	174.47	174.52
02-03-2020 19:00:00	174.47	174.49
02-03-2020 20:00:00	174.46	174.49
02-03-2020 21:00:00	174.45	174.45
02-03-2020 22:00:00	174.44	174.46

Graficamente, o resultado desta previsão, foi o seguinte:

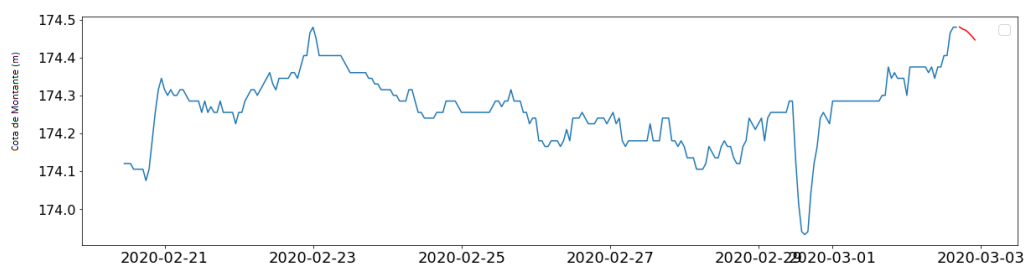


Figura 6.12: Previsão futura além do conjunto de dados existente (1)

Sendo esta a previsão resultante do modelo de estimações futuras, é possível conjugar a relação entre a Cota de Montante Real e a Prevista pelo seguinte gráfico:

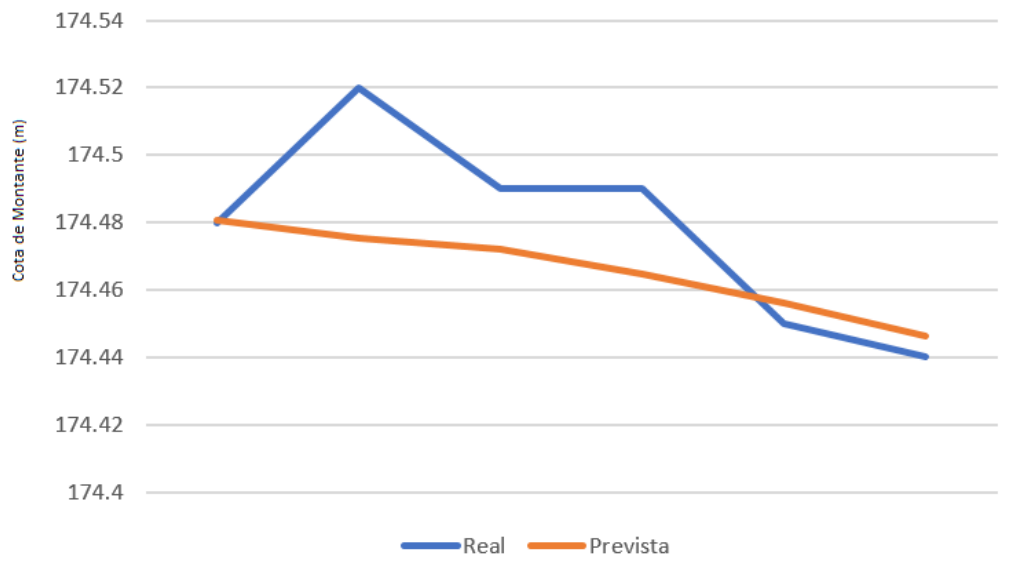


Figura 6.13: Relação, em metros, entre a Cota de Montante Real e Prevista de 02/03/2020

Mesmo não existindo 100% de concordância entre os dados reais e os previstos, é possível verificar-se que o modelo previu o declínio do nível da Cota de Montante da barragem da Bouçã para as seguintes horas. Prever com completa exatidão a variação futura do nível da Cota de Montante é praticamente impossível, no entanto, se o modelo apresentar eficácia na previsão de um padrão na variação da Cota, este pode ser bastante valioso.

Numa segunda análise ao modelo de previsões futuras, recolheu-se uma amostra de dados onde o nível da Cota de Montante encontrava-se constante, e repentinamente sofreu uma variação. Os resultados obtidos foram os seguintes:

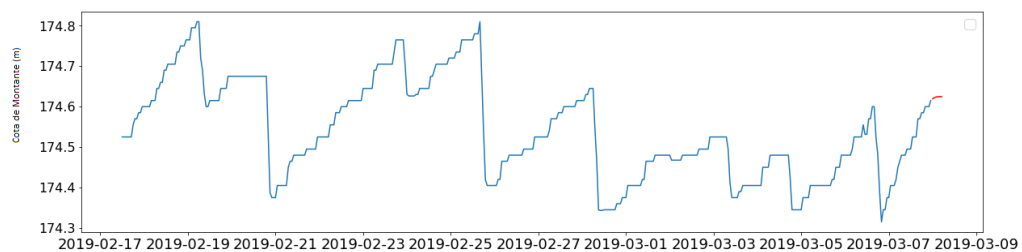


Figura 6.14: Previsão futura além do conjunto de dados existente (2)

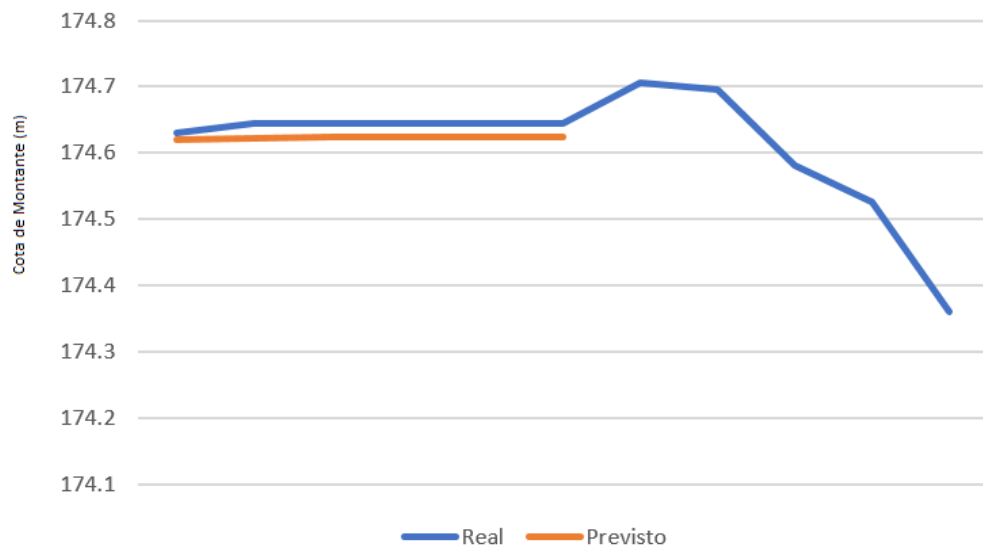


Figura 6.15: Análise da relação, em metros, entre a Cota de Montante Real e Prevista de 09/03/2019 (1)

É possível verificar-se que a previsão acompanhou muito bem o nível da Cota de Montante, no entanto, no momento que a barragem sofreu uma súbita diminuição da quantidade de água armazenada, o modelo não foi capaz de o prever:

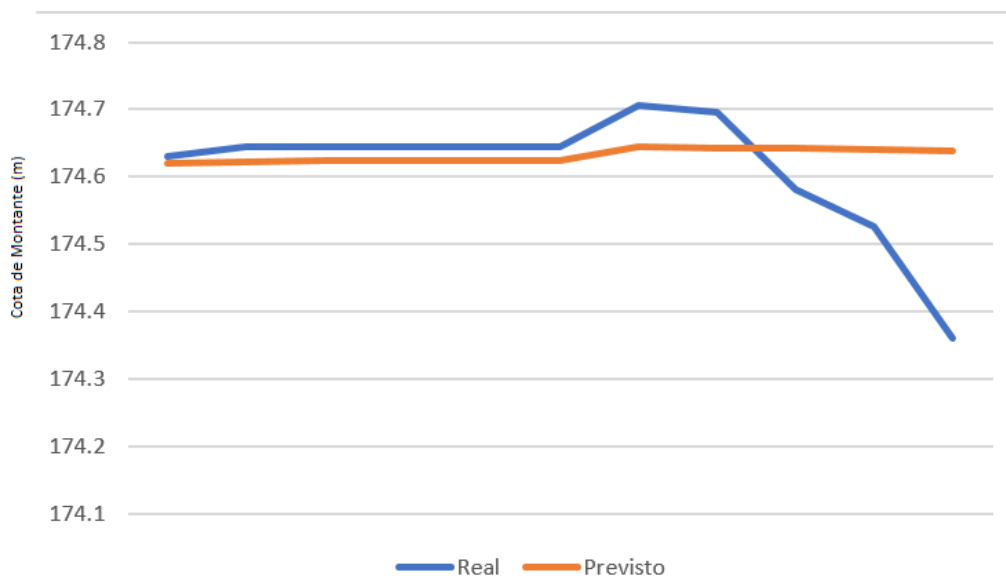


Figura 6.16: Análise da relação, em metros, entre a Cota de Montante Real e Prevista de 09/03/2019 (2)

Apesar de por um breve momento, o modelo ter apresentado uma subida do nível da Cota, este não foi capaz de prever as seguintes horas restantes.

Tabela 6.2: Resultados das previsões futuras além do *dataset* (2)

Data	Cota Prevista	Cota Real
08-03-2019 00:00:00	174.61	174.63
08-03-2019 01:00:00	174.62	174.64
08-03-2019 02:00:00	174.62	174.64
08-03-2019 03:00:00	174.62	174.64
08-03-2019 04:00:00	174.62	174.64
08-03-2019 05:00:00	174.62	174.64
08-03-2019 06:00:00	174.64	174.71
08-03-2019 07:00:00	174.64	174.69
08-03-2019 08:00:00	174.64	174.58
08-03-2019 09:00:00	174.64	174.52
08-03-2019 10:00:00	174.63	174.36
08-03-2019 11:00:00	174.63	174.39

É aceitável que tal fenómeno ocorra pois pedidos de descargas de água podem acontecer a qualquer momento, sendo impossível para o modelo prever essas situações, situadas fora do comportamento padrão natural da barragem.

6.4 Conclusões

Ao longo deste capítulo foram apresentados os diversos casos de teste, aplicados ao modelo de estimação de estados, juntamente com os seus respetivos resultados. Sendo que o principal objetivo desta fase do projeto foi a calibração do estimador, podemos concluir que tal ocorreu com sucesso.

Os resultados do modelos iniciais encontravam-se relativamente afastados do valor real da Cota de Montante, no entanto, com o ajuste das parametrizações da rede neuronal, atingiu-se um desvio quase nulo entre o valor da cota real e a prevista.

Neste típico cenário de aplicação, a inserção de um elevado número de *epochs* e um curto período de amostragem, demonstrou-se ser a técnica que melhor conjunto de resultados apresentava. Desta maneira, a versão final do Estimador de Estados foi treinado com 500 *epochs* e 3 *steps* do *dataset* por amostra.

Capítulo 7

Conclusões

Perante o trabalho desenvolvido, foi possível retirar múltiplas conclusões acerca da utilização das metodologias de *Machine Learning* para a previsão e validação de resultados.

De modo a otimizar a maneira como trabalham, cada vez mais é possível verificar-se estas tecnologias a serem utilizadas com maior frequência pelas grandes empresas. Presentemente existem inúmeras redes neuronais no mercado, cada uma delas com as suas próprias especificidades, sendo a rede LSTM a selecionada para este projeto.

Ao longo do trabalho, existiram contratemplos no que diz respeito à leitura e reestruturação do *dataset*. De modo a conseguir organizar os dados corretamente, a coluna com as datas teve de ser indexada de forma a que o algoritmo conseguisse seguir a ordem correta de acontecimentos, e também pudesse dividir o *dataset* num conjunto para o treino e outro para os testes. A criação do modelo foi algo que levou algum tempo a aperfeiçoar pois existem múltiplas formas de o fazer, no entanto, grande parte delas, ou não funcionou, ou simplesmente não apresentava os resultados desejados. Assim que a rede se encontrava de acordo com o desejado, o desafio passou por afinar os resultados do modelo, de forma adapta-lo, para retornar previsões quase idênticas às reais, é descrito no capítulo de Resultados e Validações.

No final o modelo ficou complemente desenvolvido e testado, pronto a ser implementado para a realização de estimações em tempo real. Apresentando resultados bastante positivos, consegue-se concluir que o projeto foi executado com sucesso, sendo que os seus valores são representativos ao que é expectável de ocorrer, tendo em consideração o comportamento natural do nível da Cota de Montante.

7.1 Contribuições

As principais contribuições deste projeto orientaram-se no estudo e implementação de uma rede neuronal, para a validação e previsão do nível de Cota de Montante da barragem hidroelétrica da Bouçã. É essencial garantir a qualidade dos dados que chegam ao CTCH, de modo a garantir que a operação é feita da forma mais eficiente. Outra vantagem desta implementação consiste na validação em tempo real das medidas, garantindo que eventos

fora do comum serão sempre seguidos de um alerta, para que uma intervenção ocorra sempre o mais rápido possível.

Com isto, as contribuições desta Tese são:

- Análise do estado da arte sobre implementações de previsão ou estimação de estados.
- Estudo sobre os modos de operação dos aproveitamentos hidroelétricos e as suas situações extraordinárias.
- Análise e avaliação das metodologias de *Machine Learning* existentes no mercado e verificação das características das diferentes redes neuronais.
- Estudo e escolha da *framework* adequada para a integração em questão.
- Validação e correlação dos principais indicadores de variação de cota, existentes nas barragens hidroelétricas.
- Desenvolvimento de um modelo de estimação de estados, recorrendo ao uso da metodologia e *framework* selecionada.
- Realização de múltiplas avaliações de *performance*, de modo a calibrar o modelo e tornar os seus resultados o mais *fidedignos* possível.
- Desenvolvimento de um acréscimo ao modelo de estimação, que possibilita a visualização da variação das cotas até um período adicional de 6 horas.

Este trabalho conteve um papel fulcral para a obtenção de conhecimentos internos no que toca à implementação de tecnologias de *Machine Learning*. O estudo desses conceitos, juntamente com as noções sobre os aproveitamentos hidroelétricos adquiridas, permitiu a criação de um modelo de estimação de estados robusto e conciso, que pode ser utilizado no dia-a-dia, no apoio à operação dos aproveitamentos hidroelétricos da EDP Produção.

7.2 Trabalho futuro

Como trabalho futuro deste projeto fica a integração do modelo desenvolvido no Centro de Telecomando da EDP Produção, para que sempre que novos valores cheguem ao CTCH, seja desencadeado em tempo real o processo de estimação.

Tal implementação é realizada pelo seguinte conjunto de passos:

- Avaliação das possibilidades de integração existentes. Nesta fase é necessário decidir se o modelo de estimação de estados será instalado dentro da rede interna do CTCH, onde tal comunicará diretamente com o SCADA, ou então, de modo a garantir alguma segurança, executar o modelo dentro da rede EDP, no momento em que os parâmetros são enviados para a base de dados GPinfo.
- Consoante a abordagem selecionada, estudar como será realizada a leitura dos dados provenientes das centrais.

- Definir um *threshold* de aceitação, para os valores recebidos da Cota de Montante, e conseqüentemente, de que forma devem ser enviados os alarmes para as equipas operacionais, caso tal seja necessário.
- Replicar o modelo para outras centrais e alargar a estimação de dados a outras variáveis.

Bibliografia

- [1] “Edp produção.” n/a. Acedido: 2020-02-20. [citado na p. v, 12, 28, 30, 31]
- [2] “Understanding supervised, unsupervised, and reinforcement learning.” <https://mc.ai/understanding-supervised-unsupervised-and-reinforcement-learning>. Acedido: 2020-02-02. [citado na p. v, 16]
- [3] “Machine learning crash course, part ii: Unsupervised machine learning.” <https://www.iotforall.com/machine-learning-crash-course-unsupervised-learning/>. Acedido: 2020-02-26. [citado na p. v, 16, 17]
- [4] “Rede perceptron de uma única camada.” <https://www.embarcados.com.br/rede-perceptron-de-uma-unica-camada/>. Acedido: 2020-02-28. [citado na p. v, 18]
- [5] “Redes neurais recorrentes — lstm.” <https://medium.com/@web2ajax/redes-neurais-recorrentes-lstm-b90b720dc3f6>. Acedido: 2020-03-03. [citado na p. v, 18]
- [6] “A comprehensive guide to convolutional neural networks — the eli5 way.” <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. Acedido: 2020-03-03. [citado na p. v, 19, 20]
- [7] “Yolo (you only look once).” <https://towardsdatascience.com/yolo-you-only-look-once-17f9280a47b0>. Acedido: 2020-03-03. [citado na p. v, 20, 21]
- [8] “Long short-term memory.” https://en.wikipedia.org/wiki/Long_short-term_memory. Acedido: 2020-03-03. [citado na p. v, 21]
- [9] “Mcfly: Time series classification made easy.” <https://blog.esciencecenter.nl/mcfly-time-series-classification-made-easy-e47de8d29838>. Acedido: 2020-03-26. [citado na p. v, 23]
- [10] “Worldwide autonomous vehicles market trends, analysis and forecasts 2020-2024 | alphabet, bmw, tesla, byd, ford motor, daimler.” <https://www.bccourier.com/worldwide-autonomous-vehicles-market-trends-analysis-and-forecasts-2020-2024-alphabet-bmw-tesla-byd-ford-motor-daimler>. Acedido: 2020-03-27. [citado na p. v, 24]

- [11] “Funcionamento da energia hídrica e barragens hidroeléctricas.” <https://www.portal-energia.com/funcionamento-da-energia-hidrica-barragens-hidroelectricas/>. Acedido: 2020-03-13. [citado na p. v, 28]
- [12] “Barragem da bouça volta a oferecer espetáculo natural e a levar muita gente ver a queda de água.” <https://www.mundoportugues.pt/barragem-da-bouca-volta-oferecer-espetaculo-natural-levar-muita-gente-ver-queda-agua/>. Acedido: 2020-03-18. [citado na p. v, 29]
- [13] “Spyder website.” <https://www.spyder-ide.org/>. Acedido: 2020-03-26. [citado na p. vi, 35, 36]
- [14] “Understanding lstm networks.” <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Acedido: 2020-03-30. [citado na p. vi, 41, 42, 43, 44]
- [15] “Activation functions in neural networks.” <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>. Acedido: 2020-03-31. [citado na p. vi, 43]
- [16] “A energia hidroeléctrica em portugal.” http://pt.hidroerg.pt/uploads/2/2/1/5/22154310/artigo_ael_livro_apren_25.pdf. Acedido: 2020-02-28. [citado na p. 9]
- [17] “Edp produção - estudos de caso.” n/a. Acedido: 2020-02-20. [citado na p. 10]
- [18] “Revista renováveis: Grandes hídricas.” http://www.renovaveismagazine.pt/PDF/dossier_rm25.pdf. Acedido: 2020-02-20. [citado na p. 10]
- [19] “Industria e ambiente: Gestão das albufeiras em situações de seca e cheias.” <https://www.industriaeambiente.pt/noticias/gestao-das-albufeiras-em-situacoes-de-seca-e-cheias/>. Acedido: 2020-02-24. [citado na p. 10, 11]
- [20] “Estimação estática de estados em sistemas de potência.” <http://www.labspot.ufsc.br/~simoes/assp/assp2.pdf>. Acedido: 2020-03-04. [citado na p. 13]
- [21] “5 machine learning technologies you should know.” <https://www.informationweek.com/big-data/ai-machine-learning/5-machine-learning-technologies-you-should-know/d/d-id/1332627>. Acedido: 2020-02-26. [citado na p. 15]
- [22] “An overview of the supervised machine learning methods.” https://www.researchgate.net/publication/328146111_An_overview_of_the_supervised_machine_learning_methods. Acedido: 2020-02-26. [citado na p. 15]
- [23] “O que é uma rede neuronal?.” <https://home.isr.uc.pt/~paulo/PROJ/NN95/node24.html>. Acedido: 2020-02-28. [citado na p. 17]
- [24] “Perceptrons in neural networks.” <https://medium.com/@thomascourtz/perceptrons-in-neural-networks-dc41f3e4c1b9>. Acedido: 2020-02-28. [citado na p. 18]

- [25] “Illustrated guide to lstm’s and gru’s: A step by step explanation.” <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>. Acedido: 2020-03-03. [citado na p. 21]
- [26] “What is tensorflow? the machine learning library explained.” <https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html>. Acedido: 2020-03-04. [citado na p. 22]
- [27] “Apache mxnet on aws.” <https://aws.amazon.com/mxnet/>. Acedido: 2020-03-04. [citado na p. 22]
- [28] “What is caffe – the deep learning framework.” <https://codingcompiler.com/what-is-caffe/>. Acedido: 2020-03-04. [citado na p. 22]
- [29] “Theano.” <http://deeplearning.net/software/theano/>. Acedido: 2020-03-04. [citado na p. 22]
- [30] “Regime de caudais ecológicos em portugal: Enquadramento e casos práticos.” https://www.aprh.pt/Hidroenergia2019/docs/JTdH_Pinheiro-Aqualogus.pdf. Acedido: 2020-03-14. [citado na p. 28]
- [31] “Winpython.” <https://winpython.github.io>. Acedido: 2020-03-26. [citado na p. 35]
- [32] “Mean squared error.” <https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/mean-squared-error>. Acedido: 2020-03-31. [citado na p. 45]
- [33] “Adam - latest trends in deep learning optimization.” <https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c>. Acedido: 2020-03-31. [citado na p. 45]
- [34] “A quick example of time-series prediction using long short-term memory (lstm) networks.” <https://medium.com/swlh/a-quick-example-of-time-series-forecasting-using-long-short-term-memory-lstm-networks-ddc10dc1467d>. Acedido: 2020-04-02. [citado na p. 47]

