ModelT Transfer Learning Fine Tuning Without Data Augmentation

Uma imagem com texto, captura de ecrã, Website, Página web

Descrição gerada automaticamente

**Powered By:**

**Gonçalo Ferreira, nº 2222051**

**José Delgado, nº 2222049**

# Introdução

Este documento descreve os resultados obtidos com o modelo que usa técnicas de transfer lerning, fine tunig mas sem data augmentation, reutilizamos varias técnicas que usamos ao longo do projeto, conforme descritas no documento ‘ModelT\_transferLearning\_featureExtraction\_WithoutDataAumentation’, neste documento iremos focar nas funcionalidades novas.

Os passos para realizar o fine tuning a uma rede são os seguintes:

1. Adicionar a nossa rede personalizada em cima de uma rede base já treinada
2. Congelar a rede base (vgg19)
3. Treinar a parte que adicionámos (rede personalizada)
4. Descongelar algumas camadas da rede base (vgg19)
5. Treinar os dois modelos em conjunto o que adicionámos (rede personalizada) e o modelo base(vgg19)

Para acelerar o processo reutilizamos a rede classificadora que já tínhamos treinado no modelo de transfer lerning feature extraction sem data augmentation. Com esta abordagem podemos passar à frente as etapas um, dois e três. Então começamos por importar a VGG19 e de seguida importamos o modelo pré treinado apenas com a parte classificadora. Por fim bastou apenas ligar os modelos, primeiro criamos uma camada de input com o mesmo formato da VGG19, de seguida, passamos essa entrada pelo modelo base VGG19, obtendo a saída do modelo, a saída da vgg19 será o input do modelo pré treinado, finalmente, criamos um novo modelo que combina a camada de entrada, a saída processada pela VGG19 e o modelo pré treinado.

No notebook existe uma função chamada “print\_layer\_trainable\_status”, que usamos para verificar que camadas estão definidas como treináveis e também para verificar se a união dos modelos (VGG19 + rede pré treinada (classificação)).

Em relação aos dados, dividimos o conjunto completo em três partes distintas, treino, teste e validação. Essa abordagem permitiu-nos treinar o modelo sequencialmente com cada uma das partes do conjunto de treino, avaliando o desempenho em um conjunto de validação separado após cada fase de treino. Posteriormente, os datasets de treino e de validação cada um deles foi dividido em três partes igualmente proporcionais, garantindo que cada parte representasse aproximadamente um terço do tamanho original do conjunto, realizamos esta operação com o objetivo de os treinos serem menos demorados, conseguindo obter feedback mais rápido pelo modelo.

# Resultados

Resultado 1

Primeiramente começamos por descongelar todo o block 5 da VGG19, ou seja, as últimas 5 camadas, isto faz com que os pesos sejam atualizados ao longo do processo de treino, e realizamos alguns treinos.

A seguinte figura ilustra uma representação da VGG19 com o block 5 definido como treinável. Relembrar que a camada de max pooling do block 5 não possui pesos treináveis

A diagram of a diagram of a diagram

Description automatically generated with medium confidence

A seguinte figura mostra o resultado que obtivemos após a realização de três treinos. Na figura conseguimos concluímos que o modelo não está com overfitting e que o validation accuracy está nos 88%.

A screenshot of a computer

Description automatically generated

Devido ao resultado descrito acima iremos descongelar mais camadas da VGG19 e retirar novas conclusões sobre os resultados obtidos. Se forem melhores continuamos a descongelar camadas se não paramos e o modelo fica finalizado.

Resultado 2

Como referido no resultado 1 iriamos descongelar mais camadas e está parte demostrará os resultados obtidos.

Primeiramente começamos por descongelar todo o block 4, ou seja, agora as últimas 10 camadas ficaram definidas como treináveis.

A seguinte figura ilustra uma representação da VGG19 com o block 5 e o block 4 definidos como treináveis.

A diagram of a diagram of a diagram

Description automatically generated

A seguinte figura mostra o resultado que obtivemos após a realização de três treinos. Na figura conseguimos concluímos que o modelo melhorou a validation accuracy agora com aproximadamente 91%, tendo uma melhoria aproximada de 3%. Inconvenientemente o modelo demostrou overfitting.

A screenshot of a computer code

Description automatically generated

A seguinte figura mostra que o modelo está com overfitting, estando a accuracy do treino quase nos 96% e a accuracy da validação nos 91%.

A graph of a line and a line

Description automatically generated with medium confidence

Resultado Final

Fomos ainda descongelar o block 3 e logo no primeiro treino verificamos que o modelo não conseguiu obter melhores resultados, continuando com o overfitting e a accuracy da validação mantendo-se nos 91%. Portanto não chegamos a guardar o modelo e ficamos com o que relatamos no resultado anterior.

A seguinte figura ilustra uma representação da VGG19 com o block 5, block 4 e o block 3 definidos como treináveis, ou seja, as últimas 15 camadas ajustaram os seus pesos ao longo do processo de treino.

A diagram of a diagram of a diagram

Description automatically generated

A figura abaixo ilustra o resultado final com os datasets de teste e de validação, conseguimos observar que a validation accuracy varia entre 88% e 89%