**SJK003 - Pablo Muñoz Alcaide**

In this lab we have to analyse the classification task CIFAR10, through 4 models by increasing the regularisation capabilities:

* **Model 1:** CNN model
* **Model 2:** CNN model + maxpooling
* **Model 3:** CNN model + maxpooling + batch normalization
* **Model 4:** CNN model + batch normalization + dropout + data augment.

The code for this session has not been provided because it is just a matter of copying and pasting the code into the locations mentioned in the practice.

First, we will show the results obtained in the various models by means of a table and the corresponding learning graphs

|  | **Train Accuracy** | **Validation Accuracy** | **Loss Train** | **Loss Validation** | **Number Parameters** |
| --- | --- | --- | --- | --- | --- |
| **Model 1** | 0.9050 | 0.6690 | 0.3168 | 0.9640 | 273153322 |
| **Model 2** | 0.8681 | 0.7134 | 0.4003 | 0.8462 | 4980010 |
| **Model 3** | 0.8960 | 0.7649 | 0.2908 | 0.8080 | 4987946 |
| **Model 4** | 0.9176 | 0.8693 | 0.2300 | 0.3922 | 4987946 |

And also we are going to see the learning curves of the models:

|  |  |
| --- | --- |
| **1: CNN Model** | **2: CNN Model + MaxPooling** |

|  |  |
| --- | --- |
| **3: CNN Model + MaxPooling + Batch\_Normalization** | **4: CNN Model + MaxPooling+ Batch Normalization + dropout + data augment** |

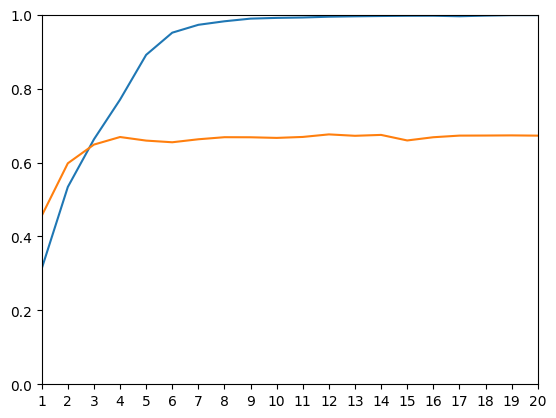
# **Model 1**

The first model is a CNN model with a series of convolutional layers, gradually increasing the number of filters (32, 64, 128, 256 and 512). Also, each convolutional layer is followed by a ReLU activation function. After the convolutional layers, the model includes a Flatten layer, which converts the multidimensional output into a one dimensional vector. Finally, the flattened output is connected to two Dense layers with 512 neurons each, where ReLU activation is used after both, and this layer is connected to the final layer which has “num\_classes = 10” neurons with a softmax activation function.   
  
The first I would like to analyse is the number of parameters of the model, which is the largest one. This may contribute to the model's ability to learn complex patterns from the data, but may increase the risk of overfitting if the dataset is not long enough, as is the case here. In addition, the training time is much higher, with 20 epochs it is up to 7 times slower than the other models with 50 epochs.

Finally, the rest of the metrics of the best model that we obtain are as follows:

* Train accuracy:0.9050
* Validation accuracy:0.6690
* Loss Train:0.3168
* Loss Validation: 0.9640

On which we can observe how well the model fits the training data (very high accuracy 0.9050, low loss function 0.3168). However, we can observe that the model does not generalise well enough, due to a much lower accuracy for new data (0.6690) and a very high value of the loss function (0.9640). Moreover, if we look at the learning curve:



We can observe more clearly the over-fitting of the model. Which converges over the epochs to an accuracy close to 100% on the training data, while it remains stagnant at around 60% accuracy for the validation set.

# Model 2

In the second model we have only added a maxpooling operator with a window

(2, 2) after each block of two convolutions that share the number of filters (including ReLU

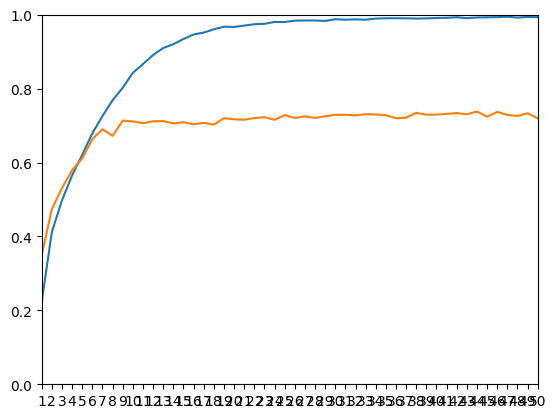
functions).

First of all, we can point out that this regularisation technique consists of dividing the input into non-overlapping regions and retaining only the maximum value from each region, that is to propagate the maximum activation value. This helps to reduce the spatial dimensions of the input, this is reflected in the number of parameters of the model (4980010). In addition, this directly affects the computational efficiency of the model, allowing training to be much faster than in the previous case.

The best model saved of that model obtain the next metrics:

* Training accuracy: 0.8681
* Validation accuracy: 0.7134
* Training Loss: 0.4003
* Validation Loss: 0.8462

Where we can see a small improvement over the previous case. Although the accuracy on training data is slightly lower and the training loss is slightly higher, the accuracy on unknown data (validation) is higher (+0.0453) and the validation loss on these data is slightly lower. This indicates that the model generalises better than the previous one and that we have slightly reduced the overfitting due to the simplification of the model by maxpooling.

Nevertheless, if we look at the learning curve over the epochs: 

As in the previous case, we see how over the epochs, the model tends to over-fit the training set, and stagnate in prediction accuracy over the validation set (although with slightly higher accuracy than the previous case).

# Model 3

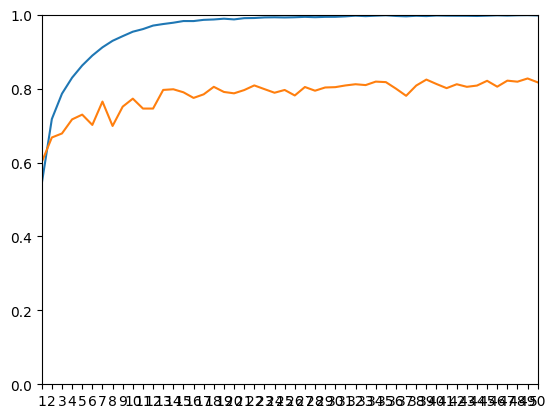
In the 3rd model we added a batch normalisation operator between each convolutional layer and its nonlinear activation function ReLU. This basically works by normalising the input of each layer in a mini-batch, centering and scaling the inputs.   
  
This regularisation technique increase a little bit the number of parameters (4987946), because this technique introduces two learnable parameters ( and ) for each feature in every layer where it is applied.

In this way, even if we have more parameters we get that the best saved model of this model generalises better, as we can see with the following metrics:

* Training accuracy: 0.8960
* Validation accuracy: 0.7649
* Training Loss: 0.2908
* Validation Loss: 0.8080

Thus, thanks to batch normalisation we avoid abrupt changes in the activation distributions, mitigating the internal covariate shift. This allows us to obtain a model that improves both the accuracy of the training model and the validation data, and the reduction of both losses. So the new model captures the unknown data much better than the previous models, i.e. we have considerably reduced the over-fitting, even though it still exists.

This smaller overfitting can be observed mainly in the learning curve:



where we can still observe stagnation in the accuracy on the validation data, while the accuracy on the training data increases steadily over the epochs.

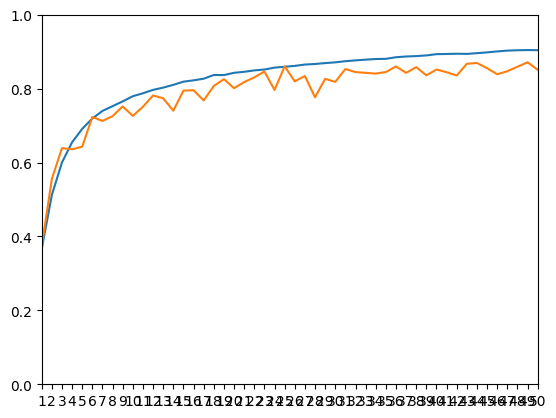
# Model 4

In the last model, we add a dropout operator after the flatten layer and after the 512-unit dense layer and we add a block of code to perform a data augmentation.

The dropout operator consists of randomly deactivating a proportion of neurons in the training phase, which helps to force the network to learn more robust features and requires less computational time. Also, the data augmentation, which is stochastic process of generating new artificial images without losing their nature, it’s a good technique which helps to avoid underfitting due to small datasets and avoid overfitting by incrementing the diversity of training data.

These techniques, together with the previous ones, allow us to obtain the following metrics of the best saved model:

* Training accuracy: 0.9176
* Validation accuracy: 0.8693
* Training Loss: 0.2300
* Validation Loss: 0.3922

Where again, we obtain a model that has increased the prediction accuracy on the training set and also significantly increased the accuracy on the validation set, as well as reducing both losses. So, although there is still a difference between the training and validation accuracies, it is very small and both accuracies are very high so we have eliminated the overfitting almost completely. This results in a model that generalises much better and predicts the unknown images very well.   
  


Finally, we can observe this null presence of overfitting in the learning curves, where we can see how throughout the epochs both precision curves grow in line and practically at the same speed.

# Conclusions

In summary, we started with a CNN model for the CIFAR10 classification task. This model had a very large number of parameters, leading to slow training and a clear presence of overfitting. Throughout the model iterations, we systematically introduced various regularisation techniques, including max pooling, batch normalisation, dropout, and data augmentation. This process involved simplifying and enriching the model, showcasing a discernible trade-off between complexity and generalisation. This iterative approach culminated in a model practically free from overfitting and with remarkably high accuracy for both known and unknown images, showing the effectiveness of these regularisation techniques.