CNNs with TensorFlow: Basics of Machine Learning

Key Takeaways

Task 1

Title: Understand the main components of neural networks in machine learning

- Neural networks consist of layers of neurons
- Each neuron receives an input vector X, a bias b, a set of weights W and returns f(W*X + b), where f is its activation function.
- The classification layer only has 10 neurons to classify pictures into 10 categories.

Task 2

Title: Train your first neural network for image classification

- The training dataset needs to be prepared before training the model.
- Images were split into training and test sets. The assigned labels were one-hot encoded to simplify the classification in the neural network.
- SGD is a good initial optimizer for a basic neural network model.
- The model looks for the values of the weights that produce the minimal loss function (measure of how far-off the predictions are from the real labels).

Task 3

Title: Improve neural network accuracy through hidden layers and different optimizers

- Hidden layers with hundreds of neurons can learn more features about the images and can classify these better than just a classification layer.
- Some optimizers for neural networks are: SGD, RMSProp, or Adam. The last two offer more robustness and flexibility than SGD, and can obtain faster convergence when training the neural network.

Task 4

Title: Visualize training data and performance of the model

- The library matplotlib.pyplot can help us to display several training images and their correct labels.

- With the same library we can also plot the training and validation accuracy as a function of the number of epochs in training.
- We can plot some misclassified images from the test set by identifying the indices in the test data where the predicted label is different from the assigned label.

Task 5

Title: Create a convolutional neural network with Conv2D and MaxPooling2D

- A convolutional neural network has a series of Conv2D and MaxPooling2D layers, followed by a Flatten layer, followed by hidden dense layers, followed by a classification layer.
- A convolutional layer performs calculations on a matrix of pixel values with a filter to identify basic spatial features like forms, shapes, and patterns in the images.
- The identification of these shapes and patterns in the pictures allow us to improve the accuracy of the neural network, even if we reduce the training epochs.

Task 6

Title: Reduce overfitting with BatchNormalization, Dropout, and L2 regularization

- L2 regularization adds a penalty on the loss function if the model gets “too complex” by having too many large weights.
- Dropout randomly drops a percentage of layer outputs to simulate training a large number of neural networks in parallel.
- BatchNormalization standardizes the inputs on the next layer to reduce overfitting.
- These measures combined allowed us to reach a 92.64% test accuracy with 50 epochs of training and the plots show no overfitting in the model.