

Assignment 7 Neural Networks with Keras and TensorFlow

Download the notebook for this assignment. The notebook will give a basis that you can use to solve the exercises.

This assignment will work with the MNIST data set. The MNIST dataset is a standard benchmark where small 28x28 pixel grayscale images of handwritten digits. For each image, the US Census Bureau manually assigned a class label, an integer from 0 to 9. The task associated with the dataset is building a **model that takes a new image (of the same size) and returns a class label – that is, an integer from 0 to 9.**

For this assignment, we will use the Keras framework to construct our neural networks. You can read more about the Keras framework here <https://keras.io/>. You can find information about the different layers and regularizers there. In this assignment, you can use free GPU leases on Google Colab or Deepnote to speed up training.

1. **Preprocessing.** In the notebook, the data is downloaded from an external server and imported into the notebook environment using the `mnist.load_data()` function call.
 - 1.1. Explain the data pre-processing highlighted in the notebook
2. **Network model, training, and changing hyper-parameters**
 - 2.1. How many layers does the network in the notebook have? How many neurons does each layer have? What activation functions and why are these appropriate for this application? What is the total number of parameters for the network? Why do the input and output layers have the dimensions they have?
 - 2.2. What loss function is used to train the network? What is the functional form (a mathematical expression) of the loss function? and how should we interpret it? Why is it appropriate for the problem at hand?
 - 2.3. Train the network for 10 epochs and plot the training and validation accuracy for each epoch.
 - 2.4. Update the model to implement a three-layer neural network where the hidden layers have 500 and 300 hidden units respectively. Train for 40 epochs. What is the best validation accuracy you can achieve? – Geoff Hinton (a co-pioneer of Deep learning) claimed this network could reach a validation accuracy of 0.9847 (<http://yann.lecun.com/exdb/mnist/>) using **weight decay** (L2 regularization of weights)

(kernels): <https://keras.io/api/layers/regularizers/>). Implement weight decay on hidden units and train and select 5 regularization factors from 0.000001 to 0.001. Train 3 replicates networks for each regularization factor. Plot the final validation accuracy with standard deviation (computed from the replicates) as a function of the regularization factor. How close do you get to Hinton's result? – If you do not get the same results, what factors may influence this? (hint: What information is not given by Hinton on the MNIST database that may influence Model training)

3. Convolutional layers

- 3.1. Design a model that makes use of at least one convolutional layer – how performant a model can you get? -- According to the MNIST database it should be possible to reach 99% accuracy on the validation data. If you choose to use any layers apart from the convolutional layers and layers that you used in previous questions, you must describe what they do. If you do not reach 99% accuracy, report your best performance, and explain your attempts and thought process.
- 3.2. Discuss the differences and potential benefits of using convolutional layers over fully connected ones for the application?

What to submit

A report that includes the figures produced and the descriptions/discussions that are requested in the questions.