

Assignment 2 group 1

By: J.N.A. Bergmans, M. Boone, J.N.A. van Mook, J.F. Peeters

Exercise 1

a) Model_1a is a model with 1 fully connected hidden layer of 64 neurons, trained with 20 epochs. A ReLu activation function was used for the hidden layer and a softmax function for the output layer. Accuracy = 0.9586 loss = 0.1427.

b) Model_1b is a model with 2 fully connected hidden layers of 64 neurons, trained with 20 epochs. A ReLu activation function was used for the hidden layers and a softmax function for the output layer. Accuracy = 0.9691 loss = 0.1042.

c) Model_1c is a model with 1 fully connected hidden layer of 128 neurons, trained with 20 epochs. A ReLu activation function was used for the hidden layer and a softmax function for the output layer. Accuracy = 0.9648 loss = 0.1209 .

d) Model_1d is a model with 2 fully connected hidden layers of 128 neurons, trained with 20 epochs. A ReLu activation function was used for the hidden layers and a softmax function for the output layer. Accuracy = 0.9712 loss = 0.0952.

e) Model_1e is a model with 6 fully connected hidden layers of 64 neurons, trained with 20 epochs. A ReLu activation function was used for the hidden layers and a softmax function for the output layer. Accuracy=0.9645 loss= 0.1239

Which model was the best was determined by looking for the model with the lowest loss and the highest accuracy for the test set. According to this method, model_1d was the best. Figure 1: The loss curves of the validation and training set of all the models in exercise 1 shows the development of the loss curves of the models. It can clearly be seen that model_1d has the lowest loss and therefore the best performance. When looking at the first four models, increasing the number of hidden layers or increasing the number of neurons per layer seems to improve model performance. However, when 6 fully connected hidden layers of 64 neurons were used, model performance was actually lower compared to the model with only 2 fully connected hidden layers of 64 neurons. It seems to be the case that when the number of hidden layers is increased too much, model performance does no longer improve.

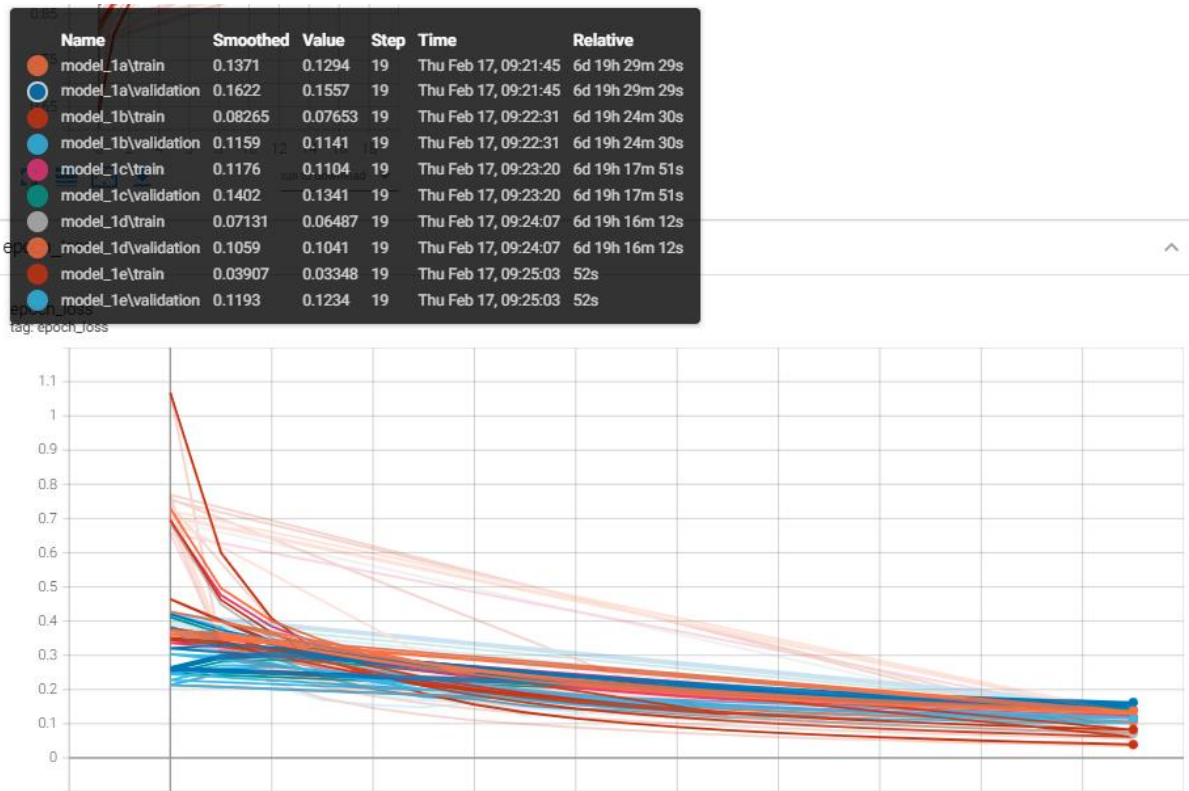


Figure 1: The loss curves of the validation and training set of all the models in exercise 1

Exercise 2

Model 2a: Neural network without any hidden layers (the input layer connects directly to the output layer)

Model 2b: Neural network with 3 hidden layers of 64 neurons with ReLU activations.

Model 2c: Neural network with 3 hidden layers of 64 neurons with linear activations (i.e. without nonlinearities between the layers).

The loss of the models on the test set is lowest for model 2b (0.0983) , highest for model 2a (0.2892) and the loss of model 2c falls between (0.2824) these two values. The model accuracy on the test set is the highest for model 2b (0.9690), the lowest for model 2a (0.9198) and model 2c's accuracy (0.9215) falls between the values for the other models. From this it can be concluded that the model 2b has the best model performance, followed by model 2c and model 2a. In Figure 2: The accuracy curves of the validation and training set of all the models in exercise 2., the development of the model training for the different models can be seen. In the figure showing the development of the loss, it can clearly be seen that model 2b has the lowest loss after 20 epochs. Furthermore, in the figure showing the development of the accuracy, it is clearly visible that model 2b has the highest accuracy after 20 epochs.

Model 2c uses a linear activation function. For a linear activation function the activation is proportional to the input. The derivative of this gradient is a constant, which makes it independent of the input. As a result, the neural network won't considerably improve the error since the gradient is exactly equal for every iteration. The neural network cannot be well-trained and capture complex data patterns.

Meanwhile, model 2b uses a ReLu activation function. This is a non-linear activation function, since for negative input values it's value will be equal to zero and for positive values it will be equal to the input value. The gradient of the ReLu function is equal to 0 for values below 0, and equal to 1 for values larger than 0. This results in the weights for some neurons being updated, while the weights and biases for other neurons are not. This way, this model can more accurately describe complex structures in the data. For this neural network classification model specifically, it means that the model will perform better at the classification task. (CS231n Convolutional Neural Networks for Visual Recognition., n.d.)

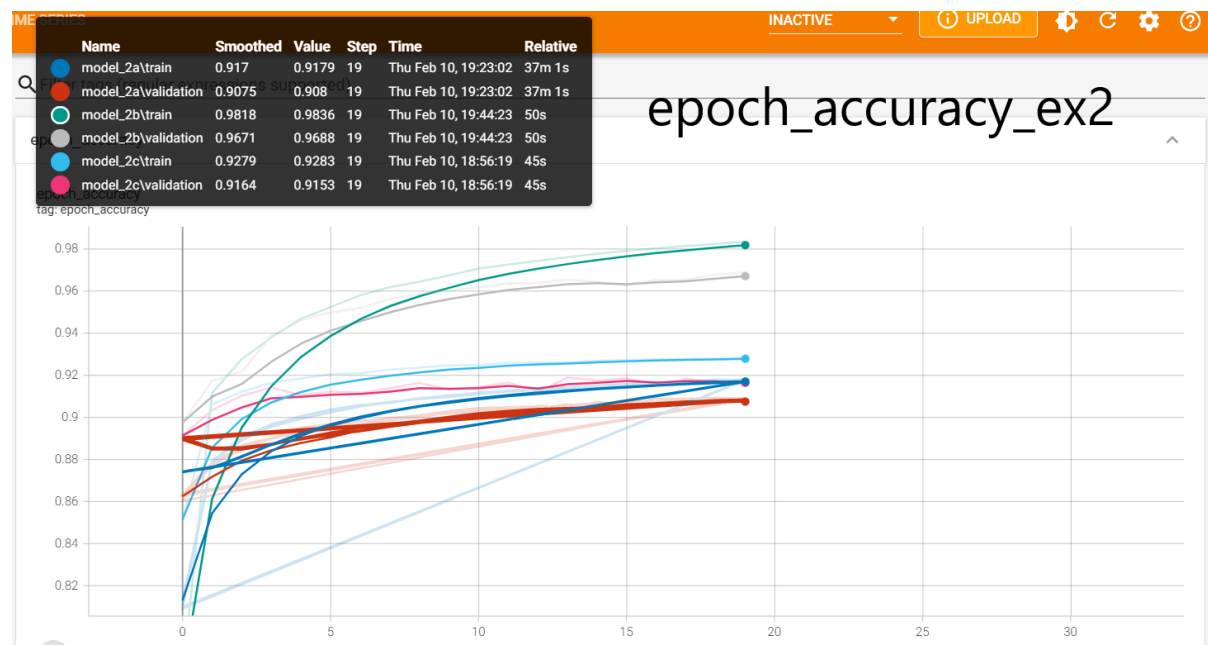


Figure 2: The accuracy curves of the validation and training set of all the models in exercise 2.

References

CS231n Convolutional Neural Networks for Visual Recognition. (sd). Opgehaald van cs231n:
<https://cs231n.github.io/neural-networks-1/#add>