Artificial Neural Networks (ANN)
Multi Layer Perceptron (MLP)

Michael Claudius, Associate Professor, Roskilde

22.04.2020, Revised 11.11.2020, 18.04.2021

Multilayer Perceptron (MLP)

- **Perceptron stand alone cannot solve many problems. Also simple binary problems: e.g. A XOR B !**
- **Solution is a layered architecture**

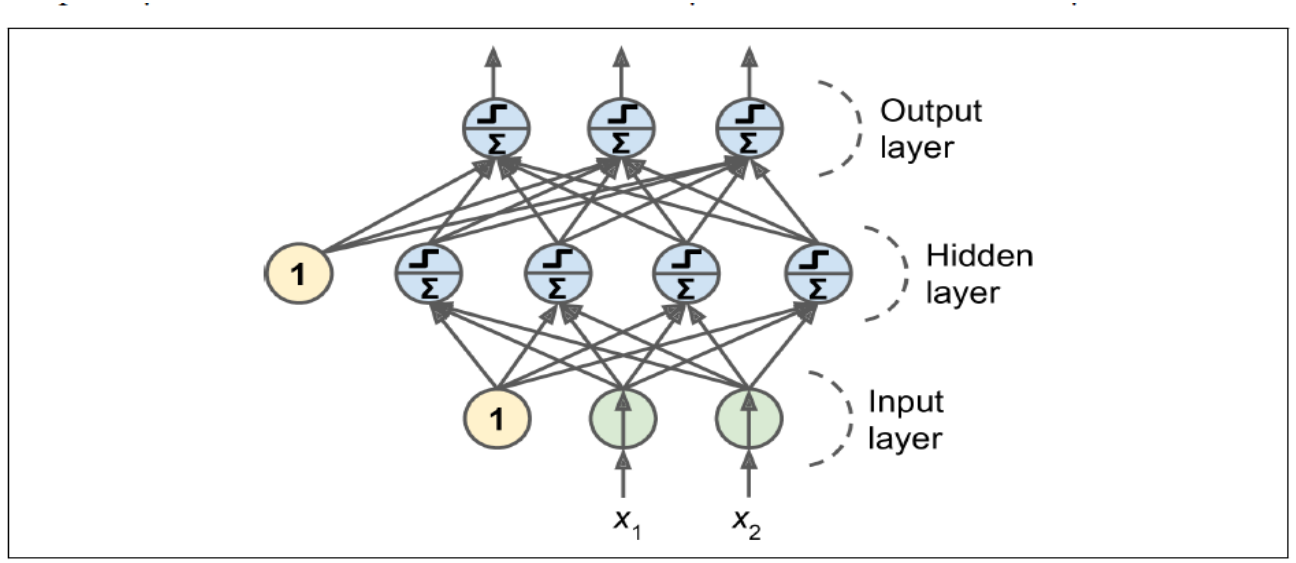

Figure 10-7. Architecture of a Multilayer Perceptron with two inputs, one hidden layer of four neurons, and three output neurons (the bias neurons are shown here, but usually they are implicit)

- **Problem: How to train? Don't worry we have backpropagation since 1986 !**

Backpropagation agorithm

- **Take a mini-batch (typically 32 instance) one at a time**
- **Forward pass (prediction)**
  - **Send mini-batch to first layer**
  - **Calculate output**
  - **Send result to next layer**
  - **Finally we have the output in output layer**
- **Calculate the output error (using a loss function)**
- **Backward pass**
  - **Measure error gradient on connection weights**
  - **Performs Gradient Descent to change all connection weights**
- **Take next mini-batch; Repeat the steps**
- **Pass the training set many times called epochs**

- **Problem: Cannot use activation step function in Gradient Descent, Need another activation function!**

Activation functions

- **Many types, 3 popular types in use**
  - **ReLU: Rectified Linear Units (maybe Softplus version)**
  - **Sigmoid, Sigmoid function from logistic regression (classification)**
  - **Tanh: Tangent hyperbolic**

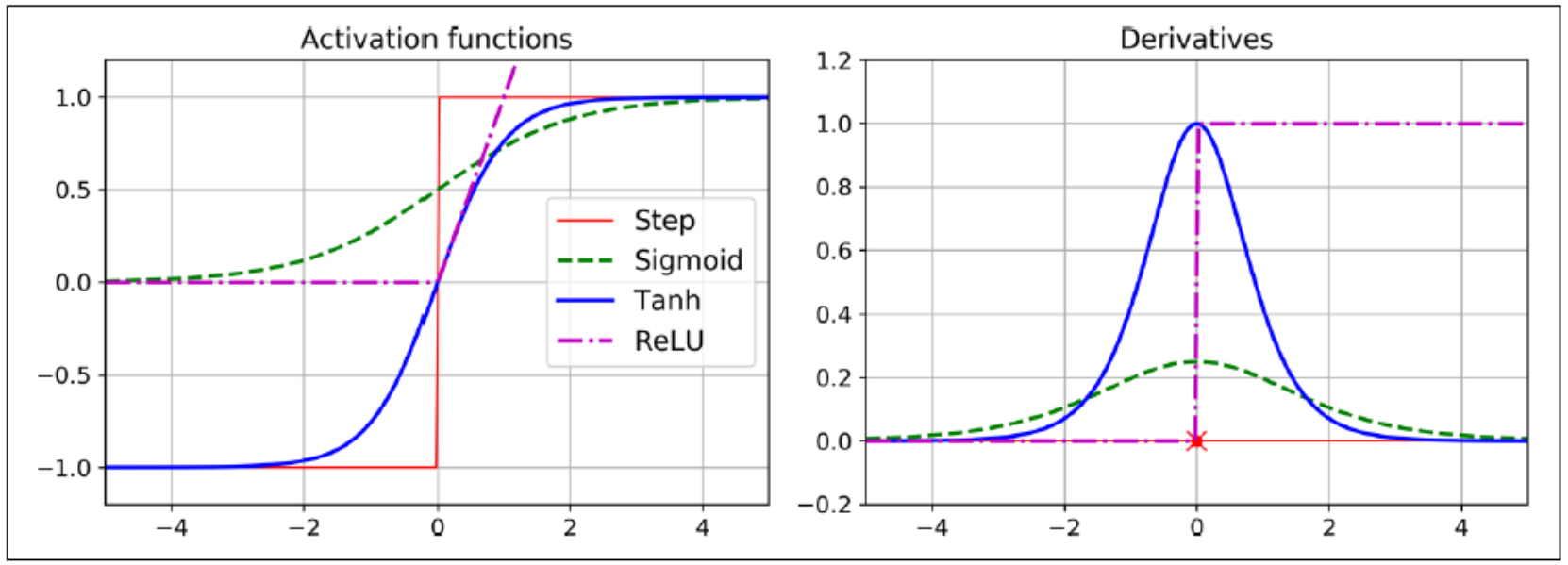

Figure 10-8. Activation functions and their derivatives

Multilayer Perceptron (MLP): Modern Classification

- **Input layer with features with bias 1**
- **One or more hidden layers with neurons, bias 1 and using ReLU: Recitfied Linear Units**
- **Output layer with Sigmoid function (probability 0-1) and one neuron per class**

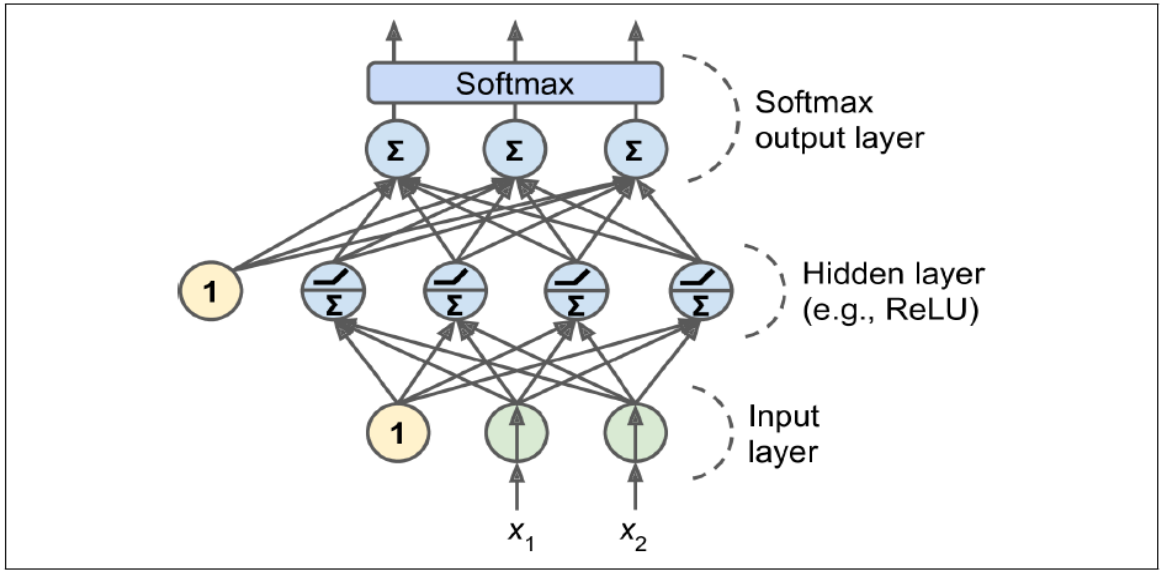

Figure 10-9. A modern MLP (including ReLU and softmax) for classification

MINST Fashion Dataset

- **10 classes: sneaker, coat, ….**
- **Feature is a 28x28 pixel picture**
- **70.000 pictures: 60.000 in training set 10.000 test set**

Figure 10-11. Samples from Fashion MNIST

Keras model: Sequential Model Code

- **Load the data**
- **Set up validation set for the epochs**
- **Flatten 2-dimensional matrix (array) to 1-dim array**
- **Create a model with two inner layers and one output layer**
- **Dense all neurons connected to next layer's neurons**

```
X_valid, X_train = X_train_full[:5000] / 255.0, X_train_full[5000:] / 255.0
y_valid, y_train = y_train_full[:5000], y_train_full[5000:]

model = keras.models.Sequential([
    keras.layers.Flatten(input_shape=[28, 28]),
    keras.layers.Dense(300, activation="relu"),
    keras.layers.Dense(100, activation="relu"),
    keras.layers.Dense(10, activation="softmax")
])
```

- **Now let's look at the whole program code**

Learning curves: Accuracy and loss

- **Accuracy: percentage correct**
- **Loss: Cross-Entropy (Chapter 4) for classification, MSE for regression**

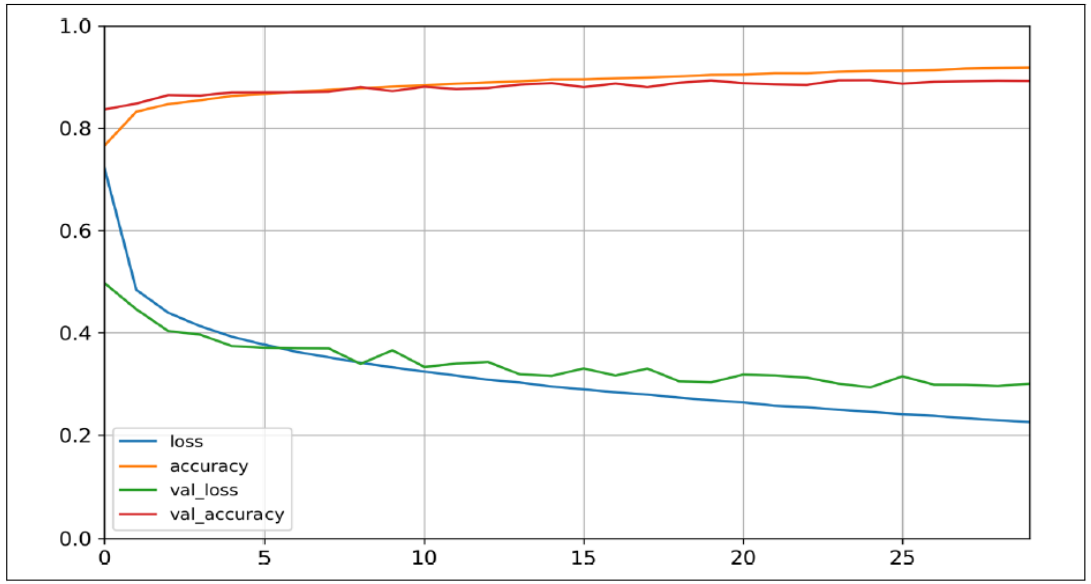

Figure 10-12. Learning curves: the mean training loss and accuracy measured over each epoch, and the mean validation loss and accuracy measured at the end of each epoch

Fine tuning hyperparameters

- **Number of layers**
- **Number of neurons**
- **Neurons distribution**
- **Activation function**

Normally more layers better than many neurons
Neurons more in start than later, Claudius' favourite: 200, 50, 50 10
ReLU is standard but you never know !

Code and Assignment!

- **First a small round tour in the code,**
- **Then more assignments**
- **Later at home, MAYBE relax with a video or two**

Assignments second round

- **It is time for discussion and solving a few exercises in groups**

- MLP Classfication Fashion Exercise
- Playground at tensorflow.org