Chapters seemed most useful to me:

Notebook 3.1 -- Shallow neural networks I

This notebook builds a shallow 1-1-3 neural network and computes its preactivations, activations, weighted activations, and final output across a 1D input range. It then varies individual θ and φ parameters to visualize how each parameter affects the resulting piecewise-linear function. Finally, it introduces training data, evaluates the network's predictions, computes the least-squares loss, and allows manual parameter tuning to reduce the error.

Notebook 3.2 -- Shallow neural networks II

This notebook constructs a shallow neural network with two inputs and three hidden units, evaluates the three linear preactivation surfaces, applies ReLU to obtain activation surfaces, and computes the weighted sums to form the final 2D output. It then visualizes these components as heatmaps to show how the network partitions the input plane into piecewise-linear regions. Finally, the notebook extends the model to produce two separate outputs by assigning two different sets of output weights and plotting each resulting output surface.

Notebook 4.3 Deep neural networks

This notebook demonstrates how shallow and deep neural networks can be expressed and evaluated in matrix form. It first computes the forward pass of a 1–1–3 network directly, then reconstructs the same model using parameter matrices to show their equivalence. It then composes two shallow networks by deriving the combined matrices that represent them as a single network. Finally, it extends the method to a deeper architecture with three hidden layers by initializing matrices of the appropriate sizes, running the full forward pass through ReLU activations, verifying dimension consistency, and reporting the resulting outputs.

Notebook 6.2 Gradient descent

This notebook builds a gradient-descent procedure for fitting a straight-line model to a small dataset. It defines the model and loss function, visualizes the loss surface, computes the gradient and verifies it with finite differences, and then iteratively updates the model parameters using gradient descent with a 1-D line search. The notebook also plots the fitted line at each step and shows the full optimization trajectory on the loss landscape.

Notebook 7.2 Backpropagation

The notebook constructs a deep neural network with five hidden layers, initializes all weight matrices and bias vectors with random values, and performs a full forward pass by computing each layer's preactivation through matrix multiplication and adding biases, followed by applying the ReLU activation. It then computes a least-squares loss against a target value and runs a complete backward pass that calculates the gradients of every weight and bias using stored forward-pass values. Finally, the notebook performs finite-difference checks on each parameter by slightly perturbing weights and biases, recomputing the loss, and comparing the numerical gradients with the backpropagated ones.