1 Definitions

1.1 Gradient descent

- gradient descent is an iterative optimization algorithm used to minimize a function
  by iteratively moving in the direction of *steepest descent* as defined by the *negative of the gradient*
- use gradient descent to update the parameters (weights w, b) of the model
- find the optimal weights that reduce the prediction error (minimize loss)

Gradient descent algorithm

- step 1: initialize the weights with random values and calculate error
- step 2: calculate the gradient (the change in error when the weights are changed by a very small amount);
- step 3: adjust the weights with their gradients; helps move the weights in the direction in which the error is minimized
- step 4: use the new weights for prediction and to calculate the new error
- step 5: repeat steps 2 to 4 untill no significant error reduction

1.2 Gradient ascent

The optimization alorithm that takes steps proportional to the *positive of the gradient*, thus approaching a local maximum of that function.

1.3 Overfitting

- when the model is trying too hard to capture the noise in the training dataset
- it models the training data too well
- it doesn't generalize well to new data
- *solution*: use regulatization

1.4 Underfitting

- the model fails to correctly model the training data
- it also doesn't generalize to new data
- *solution*: use a more complex model or a deeper model

1.5 Bias and variance

- depends on the value of an optimal error for the task at hand, which is usually close to 0% (human performance)
- look at the **error** on the *train set* to determine if you have a **bias problem**
- look at the **error difference** between the *train set* and the *test set* to determine if you have a **variance** problem
- **high bias** (*underfitting*): large train set and test set error, but similar train/test performance
- **high variance** (*overfitting*): small train set error, but large test set error
- high bias and high variance (*underfitting* and *partially overfitting*): large train set error, but even larger test set error
- low bias and low variance (model seems correct): low train set and test set error

Solutions for high bias (underfitting)

- try bigger network

- train it longer
- try some optimization algorithms
- try a different network architecture

Solutions for high variance (overfitting)

- get more data (data augmentation; e.g. rotations, flipping, zooming, distortions in images)
- try regularization
- try a different network architecture
- try early stopping

Note

- less of a trade-off between bias and variance in deep neural networks

1.6 Vanishing and exploding gradients

- when training very deep neural networks the derivatives can end up either very very big or very very small, which makes training difficult
- the derivatives might increase exponentially or decrease exponentially as a function of L (number of layers), depending on the wights initial values
- make very carefull choices when initializing the weights in order to significantly reduce this problem