Optimization: process of finding the set of parameters W that minimize the loss function.

Blindfolded hiker analogy: One analogy that you may find helpful going forward is to think of yourself as hiking on a hilly terrain with a blindfold on, and trying to reach the bottom.

Gradient: best direction along which we should change our weight vector that is mathematically guaranteed to be the direction of the steepest descend (at least in the limit as the step size goes towards zero).

- When the functions of interest take a vector of numbers instead of a single number, we call the derivatives partial derivatives, and the gradient is simply the vector of partial derivatives in each dimension.

Computing the gradient numerically with finite differences: iterates over all dimensions one by one, makes a small change h along that dimension and calculates the partial derivative of the loss function along that dimension by seeing how much the function changed.

- Additionally, in practice it often works better to compute the numeric gradient using the centered difference formula: $[f(x+h)-f(x-h)]/2h$.
- Update in negative gradient direction, since we wish our loss function to decrease, not increase.
- Small steps are likely to lead to consistent but slow progress. Large steps can lead to better progress but are more risky.

Computing the gradient analytically with Calculus: given the loss function, We can differentiate the function with respect to the weights.

Gradient descent: the procedure of repeatedly evaluating the gradient and then performing a parameter update.

Mini-batch gradient descent: approach in which the gradient is computed over batches of the training data. The gradient from a mini-batch is a good approximation of the gradient of the full objective. Therefore, much faster convergence can be achieved in practice by evaluating the mini-batch gradients to perform more frequent parameter updates.

Stochastic Gradient Descent (SGD): the mini-batches contains only a single example.