BM5702 MAKİNE ÖĞRENMESİNE GİRİŞ

Hafta 5

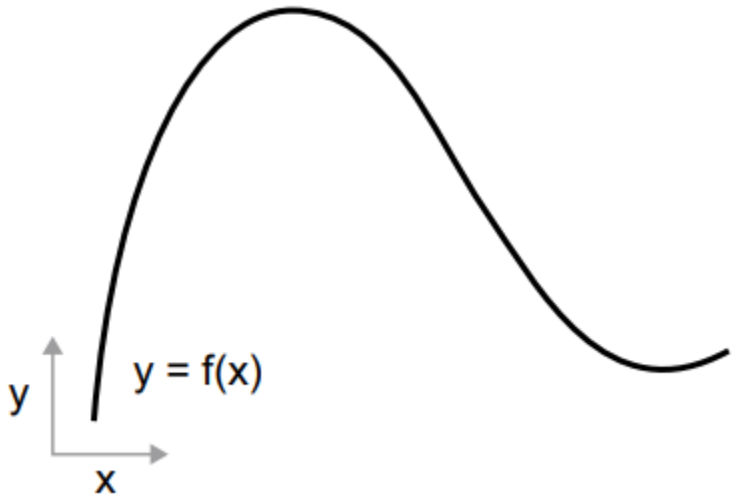
Doç. Dr. Murtaza CİCİOĞLU

Gradient-based optimization

- random initialization - gradually adjust these weights, based on a feedback signal. training loop;
  - Draw a batch of training samples, x, and corresponding targets, y_true.
  - Run the model on x (a step called the forward pass) to obtain predictions, y_pred.
  - Compute the loss of the model on the batch, a measure of the mismatch between y_pred and y_true.
  - Update all weights of the model in a way that slightly reduces the loss on this batch.

What's a derivative?

- Consider a continuous, smooth function f(x) = y, mapping a number, x, to a new number, y.
- f(x)=x^3 → f'(x)=?



Derivative of a tensor operation: The gradient

- The derivative of a tensor operation (or tensor function) is called a gradient.
  - An input vector, x (a sample in a dataset)
  - A matrix, W (the weights of a model)
  - A target, y_true (what the model should learn to associate to x)
  - A loss function, loss (meant to measure the gap between the model's current predictions and y_true)

```
y_pred = dot(W, x)
loss_value = loss(y_pred, y_true)
```

We use the model weights, W, to make a prediction for x.

We estimate how far off the prediction was.

```
loss_value = f(W)
```

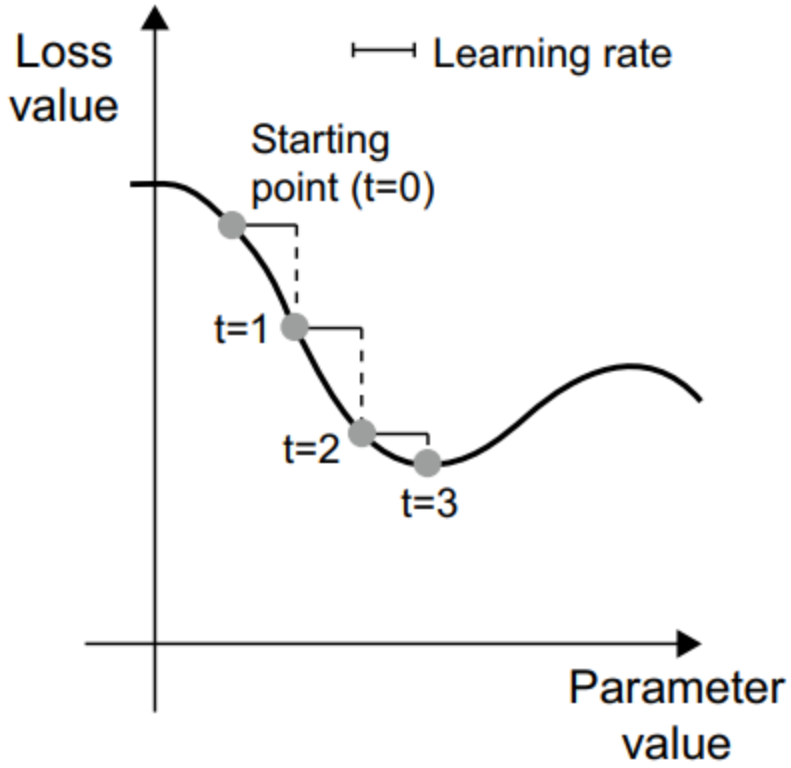
f describes the curve (or high-dimensional surface) formed by loss values when W varies.

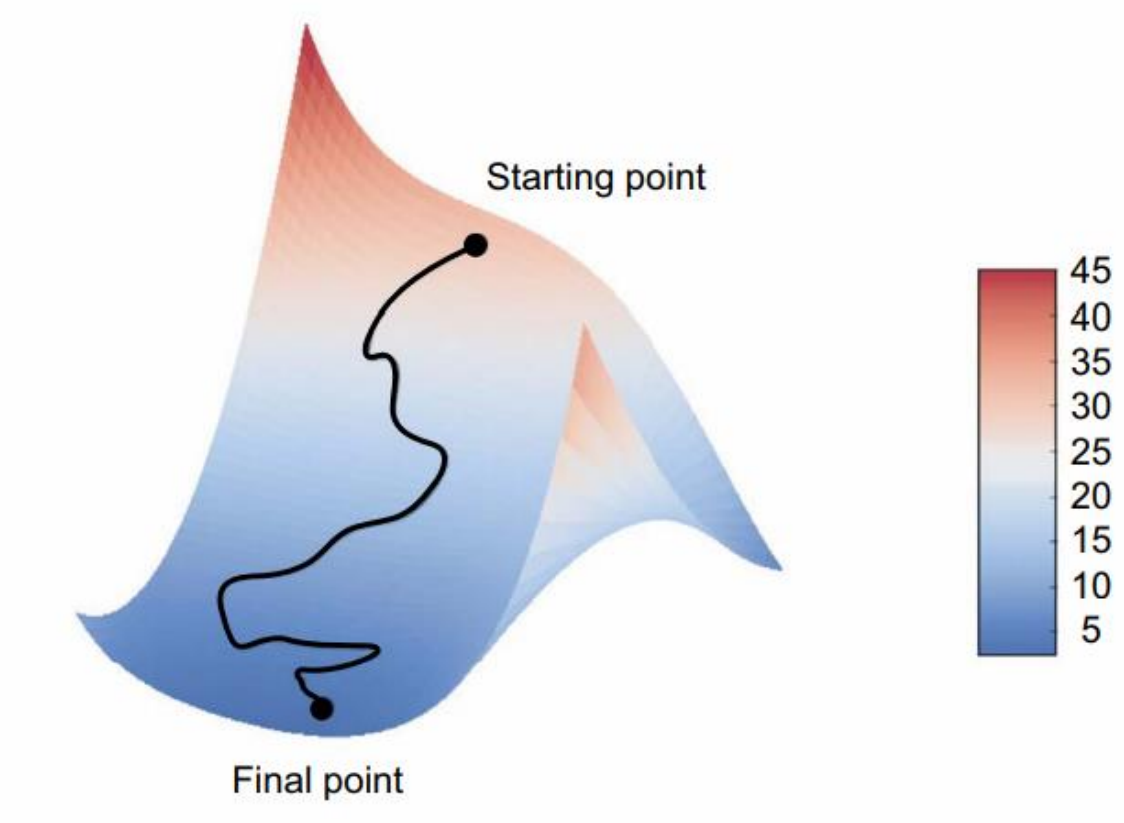
Stochastic gradient descent

- Given a differentiable function, it's theoretically possible to find its minimum analytically: it's known that a function's minimum is a point where the derivative is 0, so all you have to do is find all the points where the derivative goes to 0 and check for which of these points the function has the lowest value.
- grad(f(W), W) = 0
  - Draw a batch of training samples, x, and corresponding targets, y_true.
  - Run the model on x to obtain predictions, y_pred (this is called the forward pass).
  - Compute the loss of the model on the batch, a measure of the mismatch between y_pred and y_true.
  - Compute the gradient of the loss with regard to the model's parameters
  - Move the parameters a little in the opposite direction from the gradient
    W -= learning_rate * gradient

Stochastic gradient descent

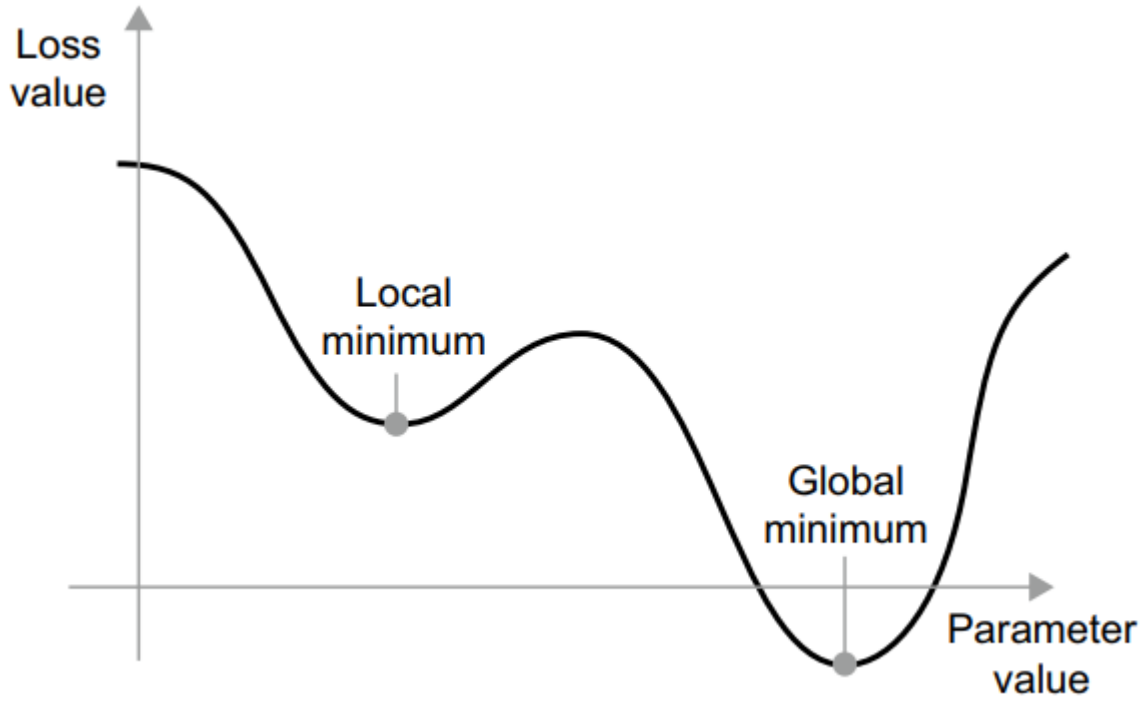
- mini-batch SGD algorithm





Stochastic gradient descent

- SGD with momentum, as well as Adagrad, RMSprop
- optimization methods or optimizers.



Chaining derivatives: The Backpropagation algorithm

- Backpropagation algorithm - THE CHAIN RULE

```
loss_value = loss(y_true, softmax(dot(relu(dot(inputs, W1) + b1), W2) + b2))
```

- Calculus tells us that such a chain of functions can be derived using the following identity, called the chain rule.
- fg(x) == f(g(x)):

```
def fg(x):
    x1 = g(x)
    y = f(x1)
    return y
```

- Then the chain rule states that grad(y, x) == grad(y, x1) * grad(x1, x).

```
def fghj(x):
    x1 = j(x)
    x2 = h(x1)
    x3 = g(x2)
    y = f(x3)
    return y
```