Deep Learning course

Edgar **Francisco** Roman-Rangel
edgar.roman@alumni.epfl.ch

Session 6 – Back-propagation

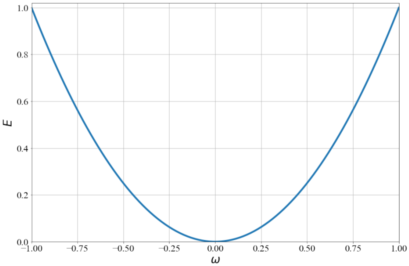
CInC-UAEM. Cuernavaca, Mexico. September 15th, 2018.

Outline

Loss function

$$\mathcal{L}(\mathbf{w}) = (h(\mathbf{x};\mathbf{w}) - y)^2$$

where $h(\mathbf{x};\mathbf{w}) = \sigma(\mathbf{w}^T\mathbf{x})$,
and $\sigma(\cdot)$ denotes a non-linear activation function.



- Navigate the loss landscape, until reaching the bottom.
- Follow the slope of the gradient $\nabla(\mathcal{L}) = \left[\frac{\partial \mathcal{L}}{\partial w_1}, \ldots, \frac{\partial \mathcal{L}}{\partial w_N}\right]^T$.

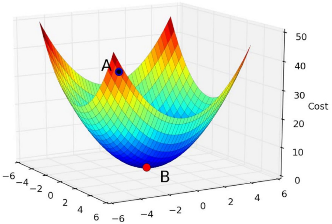
Gradient Descent (GD)

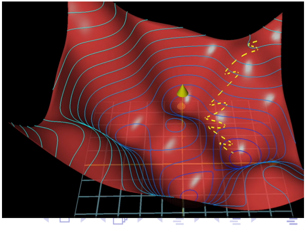
Univariate function, i.e., $\mathbf{w} = w$,

$$w = w - \eta \cdot \frac{\partial \mathcal{L}}{\partial w}$$

Multivariate function, i.e., $\mathbf{w} = [w_1, w_2, \ldots, w_N]$,

$$\mathbf{w} = \mathbf{w} - \eta \cdot \nabla(\mathcal{L})$$





Algorithm 1 Pseudocode for gradient descent.

```
Start at random location within the loss landscape.
Figure out the slope of the loss function.
repeat
    if slope > 0 then
        Move to the left.
    else if slope < 0 then
        Move to the right.
    end if
until slope = 0.
```

- The gradient gives the direction of steepest descent.
- Its sign indicates direction.
- Its magnitude indicates the importance of a particular weight.

In code:

In [7]:
```
# Training
epoch_error = list()
for epoch in range(n_epochs):
    err = list()
    for i in range(len(X)):
        x = np.insert(X[i], 0, 1)
        y_hat = predict(W, x)
        e = y[i] - y_hat
        #print(e)
        err.append(e)
        W = W + learn_rate * e * x
    print("epoch {} -- error: {}".format(epoch, err))
    epoch_error.append(np.array(err).mean())
```

Example

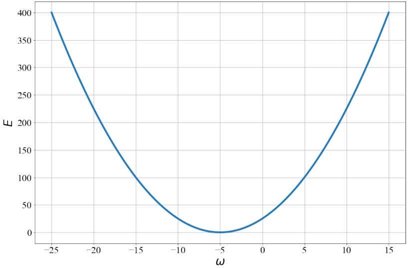
$$f(w) = (w+5)^2, \qquad \frac{df}{dw} = 2(w+5)$$

- Lets use $\eta = 0.02$,
- and say we start at $w = 3$.

Iteration 1:
$w = 3 - (0.02) * (2(3+5))$
$w = 2.68$

Iteration 2:
$w = 2.68 - (0.02) * (2(2.68+5))$
$w = 2.37$

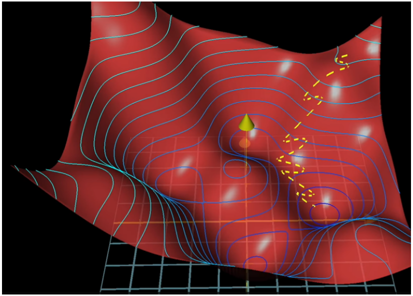
Iteration 175:
$w = -4.9934 - (0.02) * (2(-4.9934+5))$
$w = -4.9937$



Minimum when $w = -5$.

Stochastic Gradient Descent (SGD)

- For a set of pairs $\{(\mathbf{x}^{(k)}, y^{(k)})\}, \quad k = 1, \ldots, K.$
- Proceed iteratively selecting randomly one sample at a time.

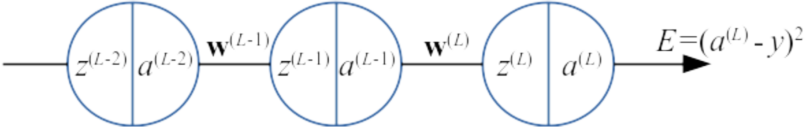


Mini batches

- SGD: multiple updates.
- Therefore, it is costly.
- Random subset (mini batch).
- Update with average.
  $$\mathbf{w} = \mathbf{w} - \eta \cdot \frac{\sum_{i=1}^{n} \nabla(\mathcal{L}_i)}{n}$$

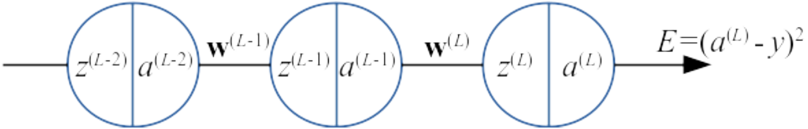
Outline

GD for deep architectures



$$\frac{\partial E}{\partial \mathbf{w}^{(L)}} = \frac{\partial E}{\partial a^{(L)}} \cdot \frac{\partial a^{(L)}}{\partial z^{(L)}} \cdot \frac{\partial z^{(L)}}{\partial \mathbf{w}^{(L)}}$$

GD for deep architectures



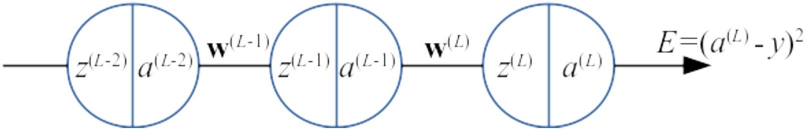
- $E = (a^{(L)} - y)^2$.
- $a^{(L)} = \sigma(z^{(L)})$.
- $z^{(L)} = \mathbf{w}^{T(L)} a^{(L-1)} + b^{(L)}$.

$$\frac{\partial E}{\partial \mathbf{w}^{(L)}} = \frac{\partial E}{\partial a^{(L)}} \cdot \frac{\partial a^{(L)}}{\partial z^{(L)}} \cdot \frac{\partial z^{(L)}}{\partial \mathbf{w}^{(L)}}$$

- $\frac{\partial E}{\partial a^{(L)}} = 2(a^{(L)} - y)$.
- $\frac{\partial a^{(L)}}{\partial z^{(L)}} = \sigma'(z^{(L)})$.
- $\frac{\partial z^{(L)}}{\partial \mathbf{w}^{(L)}} = a^{(L-1)}$.
- $\frac{\partial z^{(L)}}{\partial b^{(L)}} = 1$.

$$\frac{\partial E}{\partial \mathbf{w}^{(L)}} = 2(a^{(L)} - y) \cdot \sigma'(z^{(L)}) \cdot a^{(L-1)}$$

GD for deep architectures



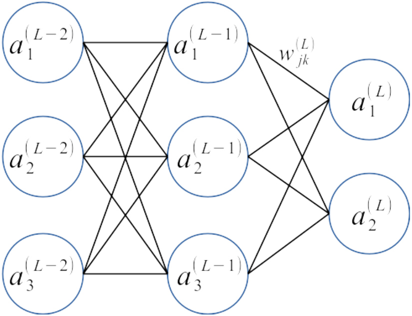
- $E = (a^{(L)} - y)^2$.
- $a^{(L)} = \sigma(z^{(L)})$.
- $z^{(L)} = \mathbf{w}^{T(L)} a^{(L-1)} + b^{(L)}$.

$$\frac{\partial E}{\partial \mathbf{w}^{(L)}} = \frac{\partial E}{\partial a^{(L)}} \cdot \frac{\partial a^{(L)}}{\partial z^{(L)}} \cdot \frac{\partial z^{(L)}}{\partial \mathbf{w}^{(L)}}$$

- $\frac{\partial E}{\partial a^{(L)}} = 2(a^{(L)} - y)$.
- $\frac{\partial a^{(L)}}{\partial z^{(L)}} = \sigma'(z^{(L)})$.
- $\frac{\partial z^{(L)}}{\partial \mathbf{w}^{(L)}} = a^{(L-1)}$.
- $\frac{\partial z^{(L)}}{\partial b^{(L)}} = 1$.

$$\frac{\partial E}{\partial b^{(L)}} = 2(a^{(L)} - y) \cdot \sigma'(z^{(L)})$$

Multiple nodes



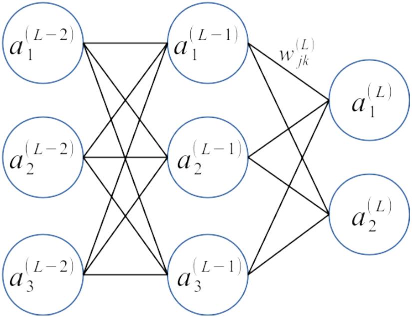
Simply incorporate relevant index for output layers.

$$\frac{\partial E}{\partial w_{jk}^{(L)}} = \frac{\partial E}{\partial a_j^{(L)}} \cdot \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \cdot \frac{\partial z_j^{(L)}}{\partial w_{jk}^{(L)}}$$

And consider all different output connections for hidden layers.

$$\frac{\partial E}{\partial a_k^{(L-1)}} = \sum_{j=1}^{n^{(L)}} \frac{\partial z_j^{(L)}}{\partial a_k^{(L-1)}} \cdot \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \cdot \frac{\partial E}{\partial a_j^{(L)}}$$

Multiple nodes



Concretely

$$\frac{\partial E}{\partial w_{jk}^{(l)}} = a_k^{(l-1)} \cdot \sigma'(z_j^{(l)}) \cdot \frac{\partial E}{\partial a_j^{(l)}}$$

where,

$$\frac{\partial E}{\partial a_j^{(l)}} = \sum_{j=1}^{n^{(l+1)}} w_{jk}^{(l+1)} \cdot \sigma'(z_j^{(l+1)}) \cdot \frac{\partial E}{\partial a_j^{(l+1}}$$

Suggested readings

- A step by step example: https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example.
- https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObC 3pi
- Chap. 6, Goodfellow, Deep Learning Book.

Thank you.

Q&A