ECE/CS 559 Lecture 9 & 10 T,Th 9/24, 9/26

Last time: Gradient Descent; Widrow-Hoff LMS Algorithm

$$\text{minimize}_w \; R(w) = \sum_{(x,y)\in Data} \ell(y, f(x;w))$$

$$w' \leftarrow w - \eta \nabla R(w) \qquad \nabla_w R = \left[\frac{\partial R}{\partial w_1}, \dots, \frac{\partial R}{\partial w_k}\right]$$

- $\ell(y, f(x;w)) = \| y - f(x;w) \|^2$

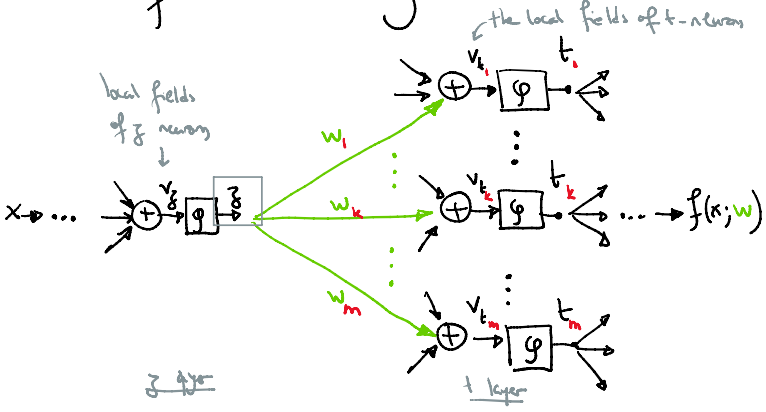
$f(x;w) = Wx$

$$W' \leftarrow W - \eta \nabla R(w) = W + 2\eta \sum_{(x,y)} (y - Wx) x^T$$

(absorb)

Back propagation

- So far we've been calculating gradients with 1 neuron.
- What if we connect many neurons?



- We will apply the chain rule multiple times:

$$\frac{\partial R}{\partial w_k} = \sum_{(x,y)} \left[\frac{\partial \ell}{\partial w_k}\right] \quad (?)$$

need [this]

(?) $\frac{\partial \ell}{\partial w_k} = \frac{\partial}{\partial w_k} \ell(t_k, \dots)\Big|_{t_k = \varphi(w_k z + \dots)}$ (v_{t_k})

have this from forward pass

(two chain rules) $= \left[\frac{\partial \ell}{\partial t_k}\Big|_{t_k=\varphi(v_{t_k})}\right] \cdot \varphi'(v_{t_k}) \cdot z$ (*)

need [this] at every neuron output

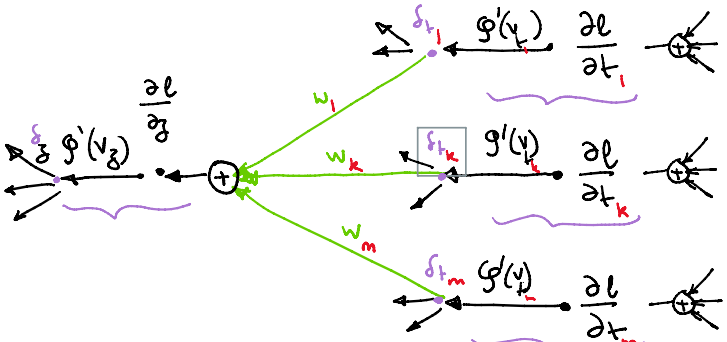
δ_{t_k}

(*) $\frac{\partial \ell}{\partial z}\Big|_{z=\varphi(u)} = \frac{\partial}{\partial z} \ell(t_1, \dots, t_k, \dots, t_m)\Big|_{t_k=\varphi(w_k z + \dots)}$ (v_{t_k})

(two chain rules) $= \sum_{k=1}^{m} \frac{\partial \ell}{\partial t_k}\Big|_{t_k=\varphi(v_{t_k})} \cdot \varphi'(v_{t_k}) \cdot w_k$

δ_{t_k}

- To calculate (?) and (*), we perform a forward pass to get all the outputs $(u, z, v_t, t, \dots)$, then we perform a backward pass:



Gradient of w = forward signal z × backward signal δ_t

- Of course, the layer of z has other (say n) neurons. Let's index the equations for these (blue j)

$$\frac{\partial \ell}{\partial w_{kj}} = \delta_{t_k} \cdot z_j \qquad \frac{\partial \ell}{\partial z_j} = \sum_{k=1}^{m} \delta_{t_k} w_{kj} \qquad j = 1, \dots, n$$

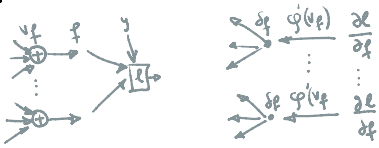
- If we think of δ_t as a m-dim vector
  of z as an n-dim vector
  of W as a $m \times n$-dim matrix
- We get the linear algebraic backward equation:

$$\nabla_W \ell = \delta_t z^T \qquad \nabla_z \ell = W^T \delta_t \qquad \delta_z = \nabla_z \ell \bullet \varphi'(v_z)$$

outer product; matrix multiplication; elementwise product

- To complete this, we need 2 things: how to start, how to handle bias
- Start at the last layer $\nabla_f \ell$ depends on the choice of ℓ

e.g. if $\ell = \|f - y\|^2$ then $\nabla_f \ell = 2(f - y)$, $\delta_f = \nabla_f \ell \bullet \varphi'(v_f)$

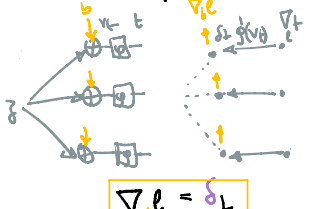


- Biases are like dead-ends on the backward path.

$$\frac{\partial \ell}{\partial b} = \frac{\partial}{\partial b} \ell(t_k, \dots)\Big|_{t_k=\varphi(\dots + b)} \quad (v_{t_k})$$

$$= \frac{\partial \ell}{\partial t}\Big|_{t_k=\varphi(v_t)} \cdot \varphi'(v_{t_k}) \cdot 1$$

δ_{t_k}



Note: Can we forward eqs to get these: $\ell(t) = \ell(\underbrace{\varphi(w_j z + b)}_{t})$ $\Rightarrow$ $\nabla_w \ell = [\underbrace{\nabla_t \ell \bullet \varphi'(v_t)}_{\delta_t}] z^T$, $\nabla_b \ell = [\underbrace{\nabla_t \ell \bullet \varphi'(v_t)}_{\delta_t}] \cdot 1$, $\nabla_z \ell = W^T [\underbrace{\nabla_t \ell \bullet \varphi(v_t)}_{\delta_t}]$