CSE 575
Statistical Machine Learning

Lecture 13

YooJung Choi

Fall 2022

Network training

- Loss function? Binary classification: cross-entropy / logistic loss

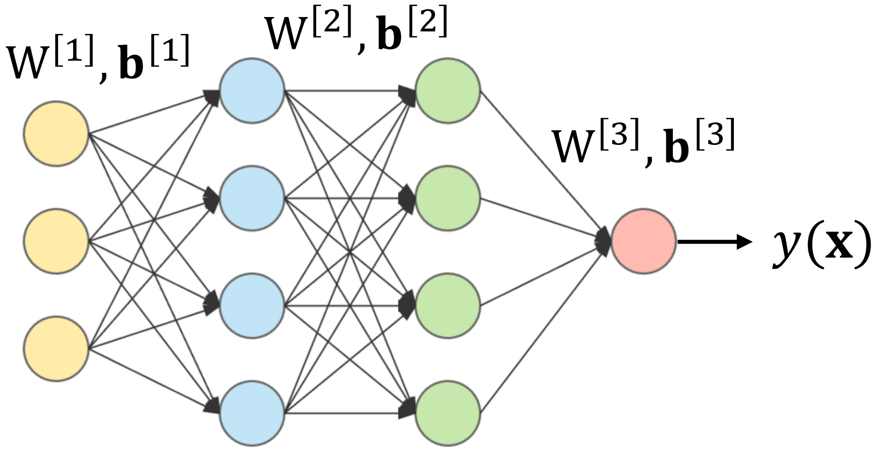
$$E(\boldsymbol{W}) = \sum_{n=1}^{N} E_n(\boldsymbol{W}) = -\sum_{n=1}^{N} \{t_n \log y(\mathbf{x}_n) + (1 - t_n)\log(1 - y(\mathbf{x}_n))\}$$

- How to optimize? Gradient descent! (or variants)

$$\boldsymbol{W}^{(new)} \leftarrow \boldsymbol{W}^{(old)} - \eta \sum_{n=1}^{N} \nabla E_n(\boldsymbol{W})$$

d dimensional inputs
w units per hidden layer (width)

$W^{[1]}$: *w* x *d*
$W^{[2]}$: *w* x *w*
$W^{[3]}$: *1* x *w*



Network training

- Computing the gradient: for every $W_{ij}^{[l]}, b_i^{[l]}$ compute $\frac{\partial E_n}{\partial W_{ij}^{[l]}}, \frac{\partial E_n}{\partial b_i^{[l]}}$

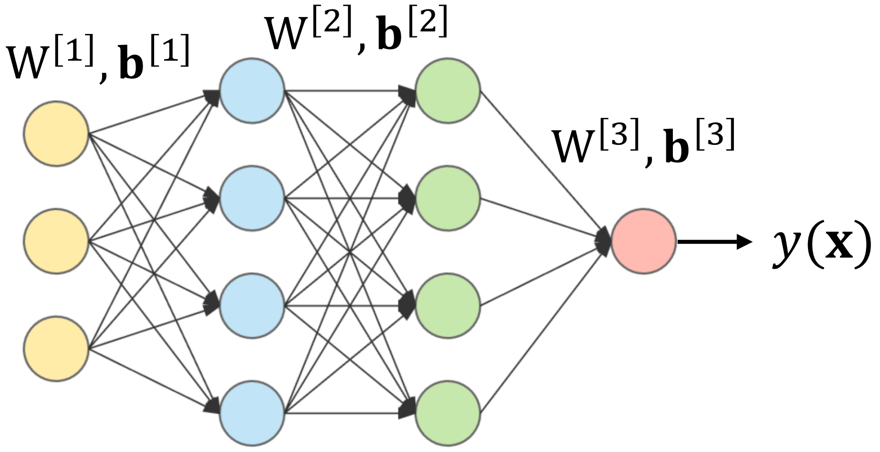
$$\frac{\partial E_n}{\partial W_{ij}^{[l]}} = \frac{\partial E_n}{\partial y(\mathbf{x}_n)} \frac{\partial y(\mathbf{x}_n)}{\partial W_{ij}^{[l]}}$$

$$y(\mathbf{x}_n) = \sigma\left(\sum\nolimits_w W_{..}^{[3]} \sigma\left(\sum\nolimits_w W_{..}^{[2]} \sigma\left(\sum\nolimits_d W_{..}^{[1]} x_{nd} + b_{..}^{[1]}\right) + b_{..}^{[2]}\right) + b_{..}^{[3]}\right)$$

Number of paths grows exponentially in network depth (number of layers)

d dimensional inputs
w units per hidden layer (width)

$W^{[1]}$: *w* x *d*
$W^{[2]}$: *w* x *w*
$W^{[3]}$: *1* x *w*



Backpropagation: example

See attached notes

Backpropagation

- In general, $\frac{\partial E}{\partial W_{ij}^{[l]}} = \frac{\partial E}{\partial a_i^{[l]}} \cdot \frac{\partial a_i^{[l]}}{\partial W_{ij}^{[l]}} = \delta_i^{[l]} \cdot z_j^{[l-1]}$

  where $\delta_i^{[l]} = \frac{\partial E}{\partial a_i^{[l]}}$ *"local gradient"*

- Output units: $\delta_i^{[l]} = \frac{\partial E}{\partial z_i^{[l]}} \cdot \frac{\partial z_i^{[l]}}{\partial a_i^{[l]}} = \frac{\partial E}{\partial z_i^{[l]}} \cdot f'(a_i^{[l]})$

- Hidden units: $\delta_i^{[l]} = \frac{\partial E}{\partial a_i^{[l]}} = \sum_k \frac{\partial E}{\partial a_k^{[l+1]}} \frac{\partial a_k^{[l+1]}}{\partial a_i^{[l]}}$

$$= \sum_k \delta_k^{[l+1]} \frac{\partial a_k^{[l+1]}}{\partial z_i^{[l]}} \frac{\partial z_i^{[l]}}{\partial a_i^{[l]}}$$

$$= f'(a_i^{[l]}) \sum_k \delta_k^{[l+1]} W_{ki}^{[l+1]}$$

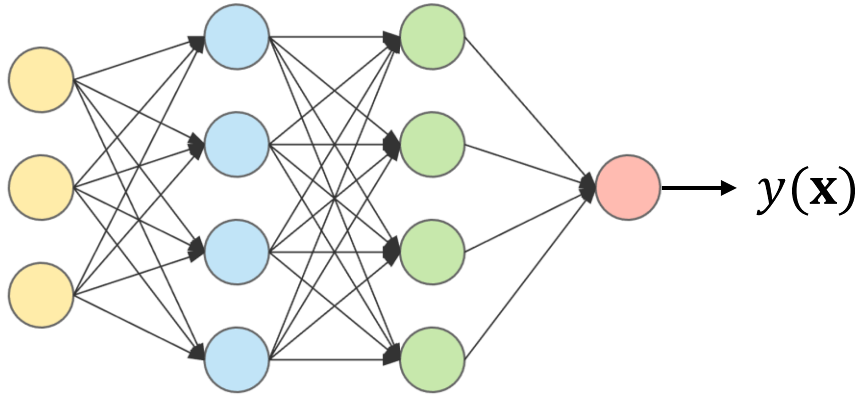
$$\boxed{\delta^{[l]} = f'\left(a^{[l]}\right)^T W^{[l+1]T} \delta^{[l+1]}}$$

Abusing notation, let $z_j^{[0]} = x_j$

Forward propagation:

$z_i^{[l]} = f\left(a_i^{[l]}\right)$

$a_i^{[l]} = \sum_j W_{ij}^{[l]} z_j^{[l-1]} + b_i^{[l]}$



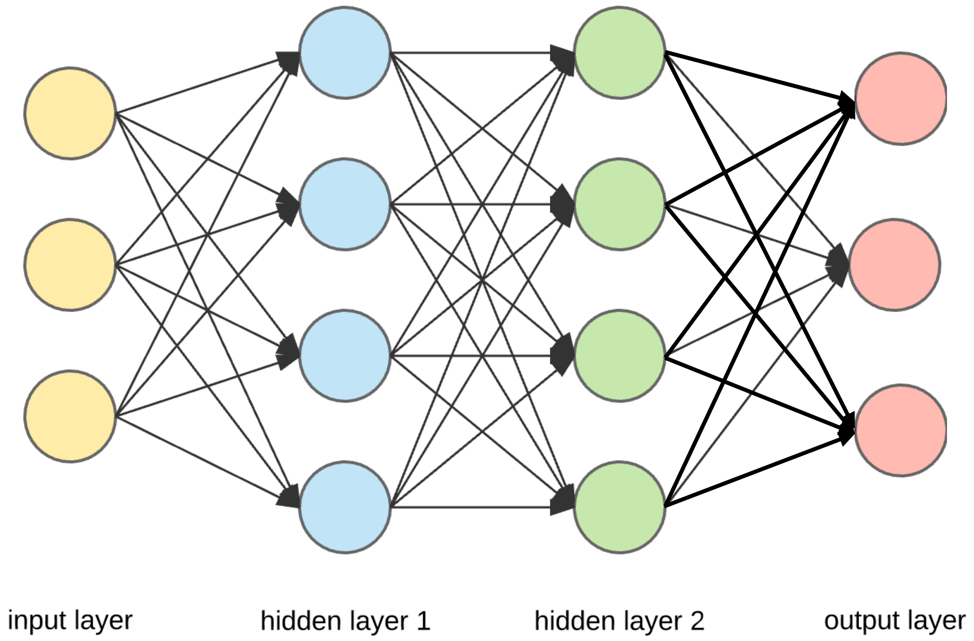
Backpropagation

1. Forward propagate an input (or batch of inputs) through the network to get $z_i^{[l]}, a_i^{[l]}$
2. Evaluate $\delta_i^{[l]}$ for all output units
3. Backpropagate to obtain $\delta_i^{[l]}$ for all hidden units
4. Evaluate the derivatives:

$$\frac{\partial E}{\partial W_{ij}^{[l]}} = \delta_i^{[l]} \cdot z_j^{[l-1]}, \qquad \frac{\partial E}{\partial b_i^{[l]}} = \delta_i^{[l]}$$

Deep learning: choices

- Output layer & error function
  - Often determined by the task & data (regression, binary classification, ...)
- Network architecture
- Hidden layer activation functions
- Improving training
  - Optimization techniques,
  - Input standardization,
  - Also influences choice of activation functions

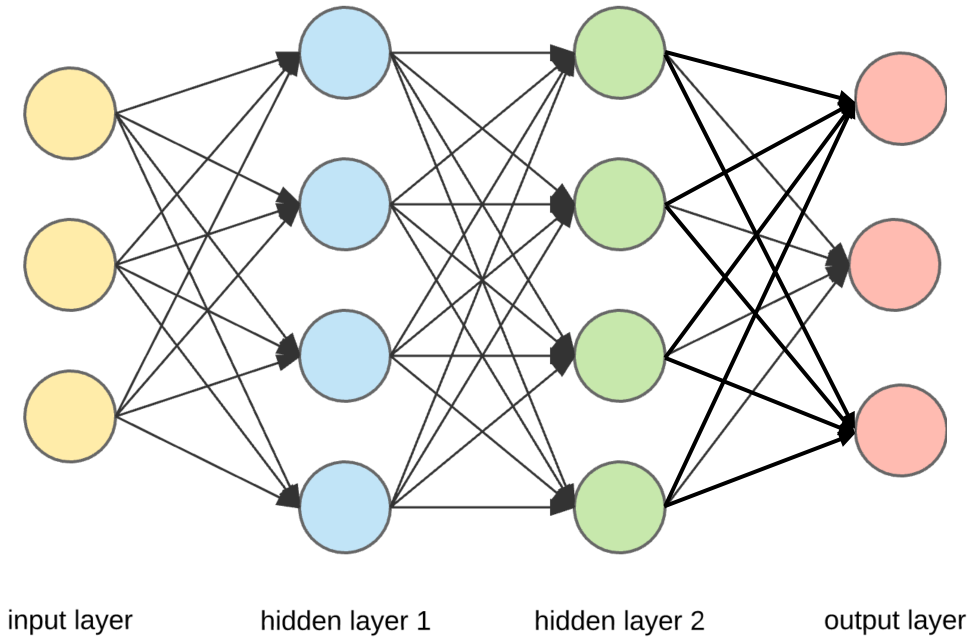


Example output units & error function

- Regression:
  - Linear/identity activation function: $y(\mathbf{x}) = z^{[L]} = a^{[L]}$
  - Squared error: $E(W, b) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - y(\mathbf{x}_n)\right)^2$
- Binary classification:
  - Logistic activation function: $y(\mathbf{x}) = z^{[L]} = \sigma\left(a^{[L]}\right)$
  - Cross-entropy (logistic) error: $E(W, b) = -\sum_{n=1}^{N}\{t_n \log y(\mathbf{x}_n) + (1 - t_n)\log(1 - y(\mathbf{x}_n))\}$
- Multiclass classification: *K output units, one for each class*
  - Softmax activation function: $y_k(\mathbf{x}) = z_k^{[L]} = s\left(a_k^{[L]}\right) = \frac{\exp\left(a_k^{[L]}\right)}{\sum_j \exp\left(a_j^{[L]}\right)}$ *Output units depend on each other!*
  - Cross-entropy error: $E(W, b) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \log y_k(\mathbf{x}_n)$

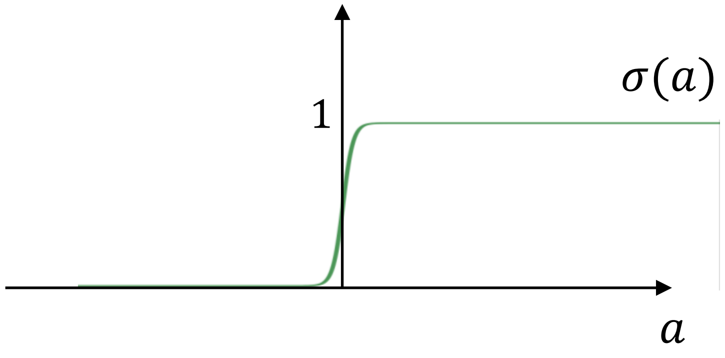
Deep learning: choices

- Output layer & error/loss function
  - Often determined by the task & data (regression, binary classification, ...)
- Network architecture
- Hidden layer activation functions
- Improving training
  - Optimization techniques,
  - Input standardization,
  - Also influences choice of activation functions



Hidden layer activation functions

- So far, we considered the sigmoid activation for hidden layer units



$$\sigma'(a) = \sigma(a)(1 - \sigma(a)) \leq (0.5)(0.5) = 0.25$$

$$\begin{aligned}
\frac{\partial E}{\partial W^{[1]}_{\cdot\cdot}} &= \delta^{[1]}_{\cdot\cdot} \cdot z^{[0]}_{\cdot\cdot} \\
&= \left(\sigma'(a^{[1]}_{\cdot\cdot}) \sum_k \delta^{[2]}_k W^{[2]}_{\cdot\cdot}\right) \cdot z^{[0]}_{\cdot\cdot} \\
&= \left(\sigma'(a^{[1]}_{\cdot\cdot}) \sum_k \left(\sigma'(a^{[2]}_{\cdot\cdot}) \sum_k \delta^{[3]}_k W^{[3]}_{\cdot\cdot}\right) W^{[2]}_{\cdot\cdot}\right) \cdot z^{[0]}_{\cdot\cdot}
\end{aligned}$$

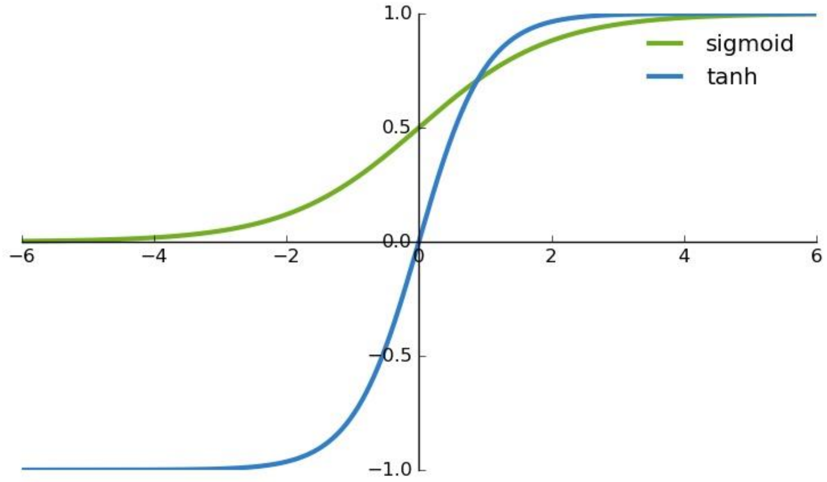
Vanishing gradient problem

- The deeper the network, the smaller the gradients become in the early layers
- i.e. harder to train the early layers (gradient descent will make very small updates)

Hidden layer activation functions

- Hyperbolic tangent: $\tanh(a) = \dfrac{\exp(a) - \exp(-a)}{\exp(a) + \exp(-a)}$

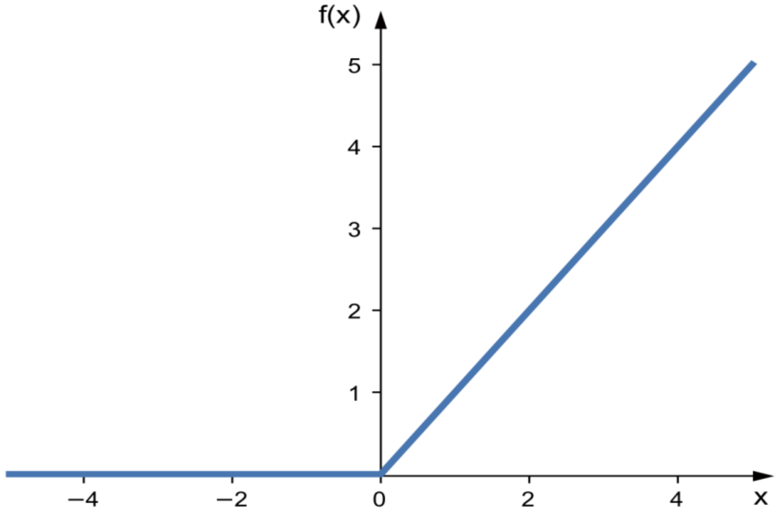
$$\tanh'(a) = 1 - (\tanh(a))^2$$

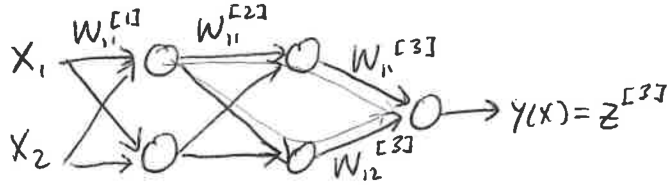


- Rectified linear unit (ReLU)

$$\mathrm{ReLU}(a) = \begin{cases} 0 & \text{if } a \leq 0 \\ a & \text{if } a > 0 \end{cases} = \max\{0, a\}$$

Most widely used activation function





$$y(x) = z^{[3]} = \sigma(a^{[3]})$$

$$a^{[3]} = W_{11}^{[3]} z_1^{[2]} + W_{12}^{[3]} z_2^{[2]} + b^{[3]}$$

$$z_1^{[2]} = \sigma(a_1^{[2]}), \quad z_2^{[2]} = \sigma(a_2^{[2]})$$

$$a_1^{[2]} = W_{11}^{[2]} z_1^{[1]} + W_{12}^{[2]} z_2^{[1]} + b_1^{[2]}$$

$$a_2^{[2]} = W_{21}^{[2]} z_1^{[1]} + W_{22}^{[2]} z_2^{[1]} + b_2^{[2]}$$

$$\left[\begin{array}{l} z_1^{[1]} = \sigma(a_1^{[1]}), \quad z_2^{[1]} = \sigma(a_2^{[1]}) \\ a_1^{[1]} = W_{11}^{[1]} x_1 + W_{12}^{[1]} x_2 + b_1^{[1]} \\ a_2^{[1]} = W_{21}^{[1]} x_1 + W_{22}^{[1]} x_2 + b_2^{[1]} \end{array}\right.$$

Given (x, t), $E(W, b) = -t \log y(x) - (1-t)\log(1 - y(x))$

$$\boxed{\frac{\partial E}{\partial W_{11}^{[3]}} = \frac{\partial E}{\partial \underset{z^{[3]}}{y}} \cdot \cancel{\frac{\partial z^{[3]}}{\partial W_{11}^{[3]}}} \; \frac{\partial z^{[3]}}{\partial a^{[3]}} \cdot \frac{\partial a^{[3]}}{\partial W_{11}^{[3]}}}$$

$$\frac{\partial E}{\partial z^{[3]}} = -\frac{t}{z^{[3]}} + \frac{(1-t)}{1 - z^{[3]}}$$

$$\frac{\partial z^{[3]}}{\partial a^{[3]}} = \sigma'(a) = \frac{\partial}{\partial a}\left(\frac{1}{1+e^{-a}}\right) = \cancel{-} \frac{1}{(1+e^{-a})^2} \cdot e^{-a} \cdot \cancel{(-1)}$$

$$= \frac{1}{1+e^{-a}} \cdot \frac{e^{-a}}{1+e^{-a}} = \sigma(a^{[3]})(1 - \sigma(a^{[3]})) = z^{[3]}(1 - z^{[3]})$$

$$\frac{\partial a^{[3]}}{\partial W_{11}^{[3]}} = z_1^{[2]}$$

$$\therefore \frac{\partial E}{\partial W_{11}^{[3]}} = \left(-\frac{t}{z^{[3]}} + \frac{(1-t)}{1 - z^{[3]}}\right)\left(z^{[3]}(1 - z^{[3]})\right) \cdot z_1^{[2]}$$

$$= (-t(1 - z^{[3]}) + (1-t) z^{[3]}) z_1^{[2]}$$

$$\boxed{= (z^{[3]} - t) z_1^{[2]}}$$

$$\frac{\partial E}{\partial W_{11}^{[2]}} = \frac{\partial E}{\partial z^{[3]}} \cdot \frac{\partial z^{[3]}}{\partial a^{[3]}} \cdot \frac{\partial a^{[3]}}{\partial W_{11}^{[2]}}$$

$$= \underbrace{\boxed{\frac{\partial E}{\partial z^{[3]}} \cdot \frac{\partial z^{[3]}}{\partial a^{[3]}}}}_{\text{from } \frac{\partial E}{\partial W_{11}^{[3]}}} \cdot \underbrace{\frac{\partial a^{[3]}}{\partial z_1^{[2]}}}_{W_{11}^{[3]}} \cdot \underbrace{\frac{\partial z_1^{[2]}}{\partial a_1^{[2]}}}_{z_1^{[2]}(1-z_1^{[2]})} \cdot \underbrace{\frac{\partial a_1^{[2]}}{\partial W_{11}^{[2]}}}_{z_1^{[1]}}$$

$$\frac{\partial E}{\partial W_{11}^{[1]}} = \frac{\partial E}{\partial z^{[3]}} \cdot \frac{\partial z^{[3]}}{\partial a^{[3]}} \cdot \frac{\partial a^{[3]}}{\partial z_1^{[1]}} \cdot \frac{\partial z_1^{[1]}}{\partial W_{11}^{[1]}}$$

$$\frac{\partial z_1^{[1]}}{\partial W_{11}^{[1]}} = \frac{\partial z_1^{[1]}}{\partial a_1^{[1]}} \cdot \frac{\partial a_1^{[1]}}{\partial W_{11}^{[1]}} = z_1^{[1]}(1-z_1^{[1]})\,x_1$$

$$\frac{\partial a^{[3]}}{\partial z_1^{[1]}} = \overbrace{\boxed{\frac{\partial a^{[3]}}{\partial z_1^{[2]}} \cdot \frac{\partial z_1^{[2]}}{\partial a_1^{[2]}}}}^{\text{from } \frac{\partial E}{\partial W_{11}^{[2]}}} \cdot \underbrace{\frac{\partial a_1^{[2]}}{\partial z_1^{[1]}}}_{W_{11}^{[2]}} + \overbrace{\boxed{\frac{\partial a^{[3]}}{\partial z_2^{[2]}} \cdot \frac{\partial z_2^{[2]}}{\partial a_2^{[2]}}}}^{\text{from } \frac{\partial E}{\partial W_{21}^{[2]}}} \cdot \underbrace{\frac{\partial a_2^{[2]}}{\partial z_1^{[1]}}}_{W_{21}^{[2]}}$$

$$\therefore \frac{\partial E}{\partial W_{11}^{[1]}} = \underbrace{\frac{\partial E}{\partial z^{[3]}} \cdot \frac{\partial z^{[3]}}{\partial a^{[3]}}}_{\text{from layer 2}} \left(W_{11}^{[2]} \underbrace{\frac{\partial a^{[3]}}{\partial z_1^{[2]}} \cdot \frac{\partial z_1^{[2]}}{\partial a_1^{[2]}}}_{} + W_{21}^{[2]} \underbrace{\frac{\partial a^{[3]}}{\partial z_2^{[2]}} \cdot \frac{\partial z_2^{[2]}}{\partial a_2^{[2]}}}_{} \right) z_1^{[1]}(1-z_1^{[1]})\,x_1$$

$$\frac{\partial E}{\partial W_{12}^{[1]}} = \frac{\partial E}{\partial a_1^{[1]}} \cdot \frac{\partial a_1^{[1]}}{\partial W_{12}^{[1]}} = \underline{\qquad\qquad\qquad} \cdot x_2$$