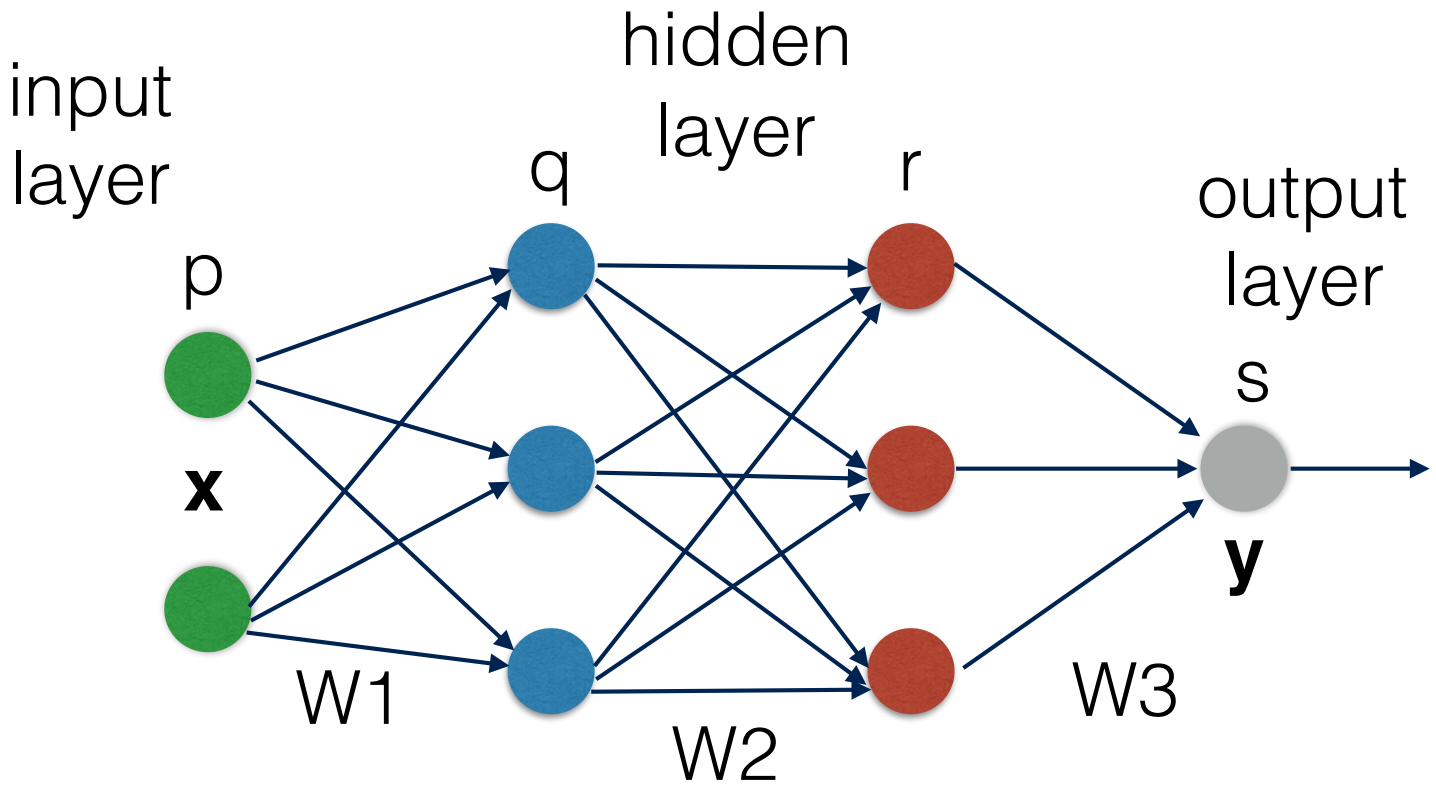
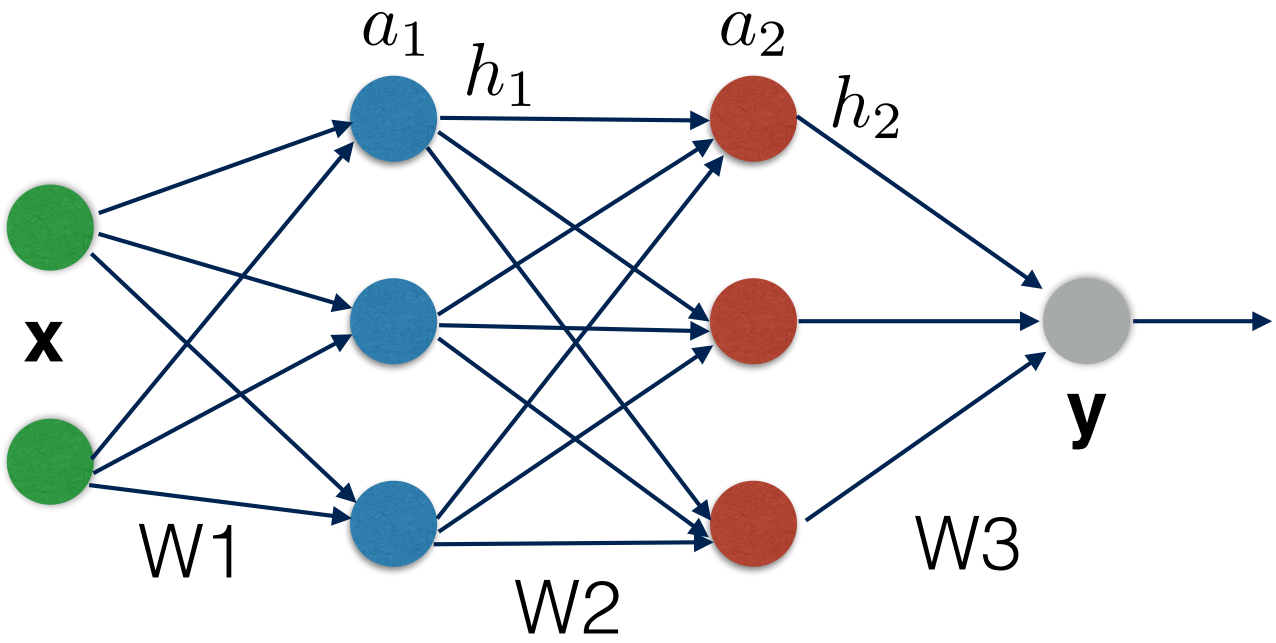
Neural Network Learning




```
W1: q x p
W2: r x q
W3: s x r
```

```
FeedForward: W3*(W2*(W1*x))
```

```
 (s x r)*((r x q)*((q x p)*(p x 1)))
=(s x r)*((r x q)*(q x 1) )
=(s x r)*(r x 1)
=(s x 1)
```



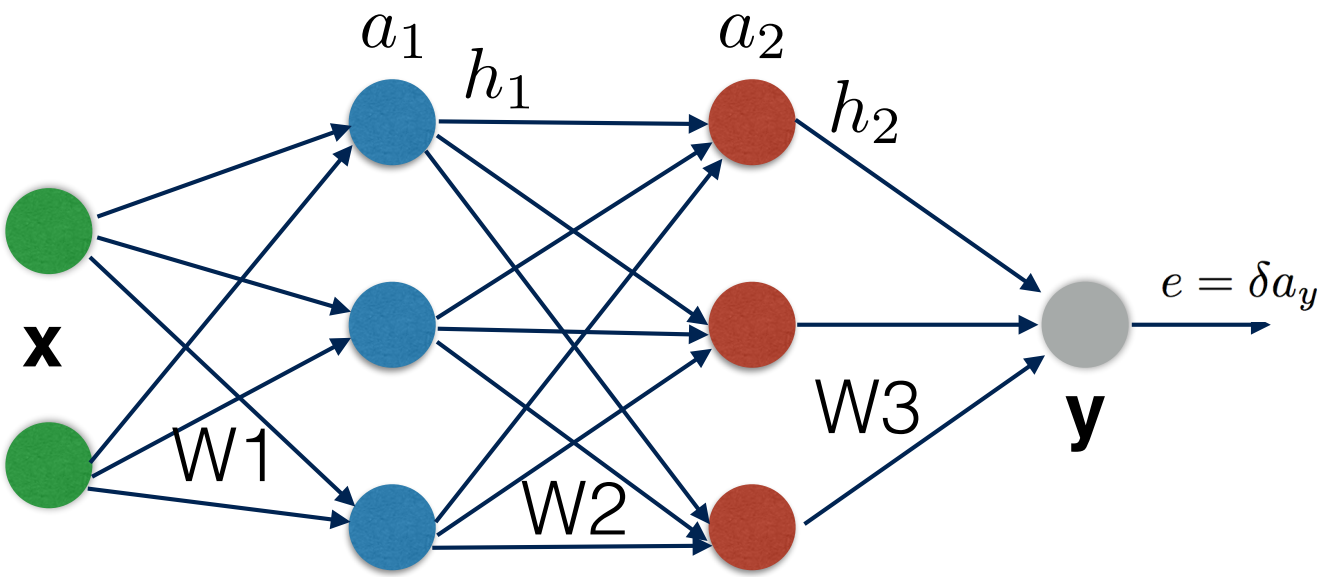
$$a_1 = W_1 x + b_1; \quad h_1 = f(a_1)$$

$$a_2 = W_2 h_1 + b_2; \quad h_2 = f(a_2)$$

$$a_y = W_3 h_2 + b_3; \quad \hat{y} = f_y(a_y)$$

$$J = -\frac{1}{N} \sum_{m,n} y_{mn} \log \hat{y}_{mn} + (1 - y_{mn}) \log(1 - \hat{y}_{mn})$$

$$e = \delta a_y = \frac{\partial J}{\partial a_y} = \hat{y} - y$$



BP

$$\delta a_2 = \frac{\partial J}{\partial a_2} = (W_3^T e) \odot f'(a_2),\ \delta a_1 = \frac{\partial J}{\partial a_1} = (W_2^T \delta a_2) \odot f'(a_1)$$

FA

$$\delta a_2 = (B_2 e) \odot f'(a_2),\ \delta a_1 = (B_1 \delta a_2) \odot f'(a_1)$$

DFA

$$\delta a_2 = (B_2 e) \odot f'(a_2),\ \delta a_1 = (B_1 e) \odot f'(a_1)$$

IFA

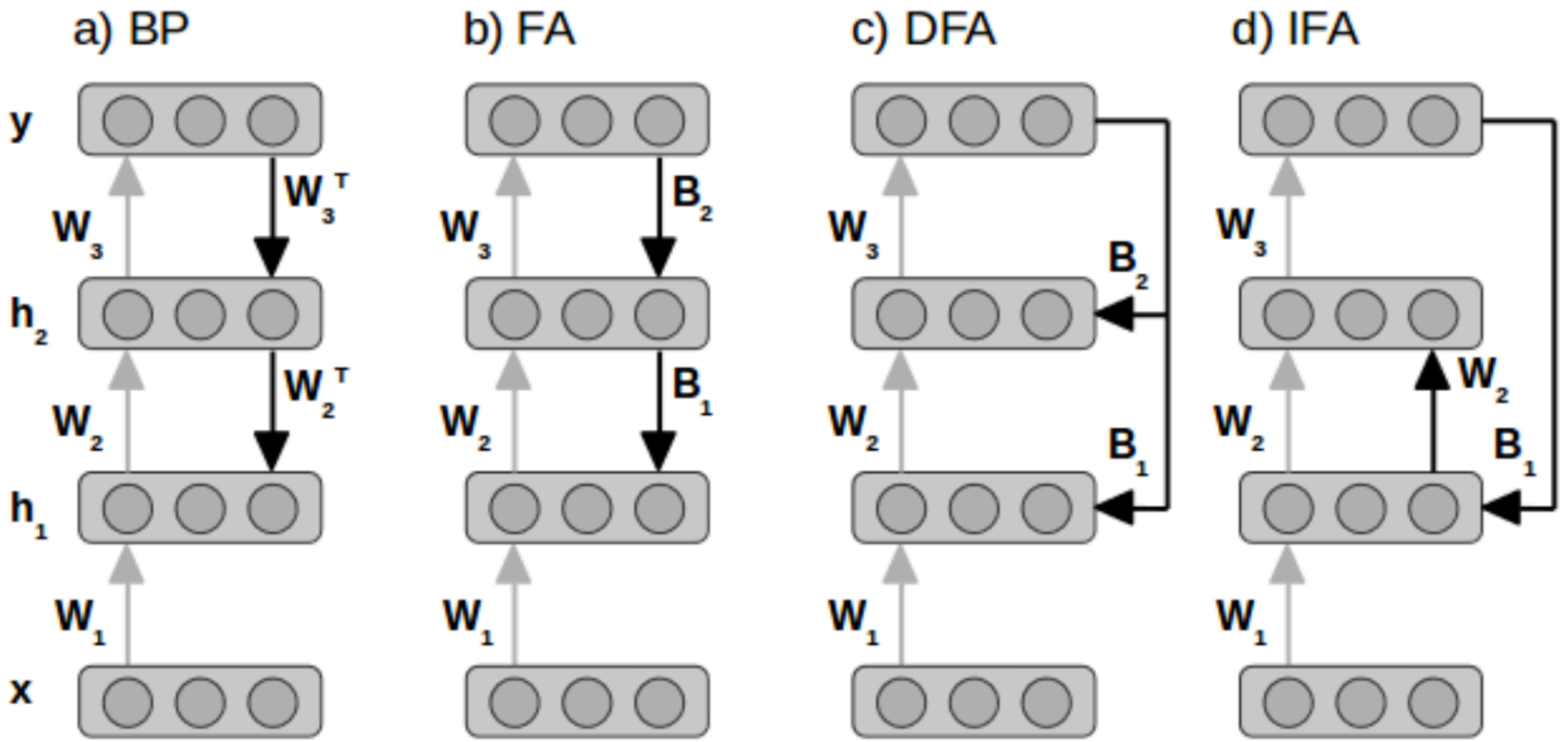
$$\delta a_2 = (W_2 \delta a_1) \odot f'(a_2),\ \delta a_1 = (B_1 e) \odot f'(a_1)$$

Weight Updates

$$\delta W_1 = -\delta a_1 x^T,\ \delta W_2 = -\delta a_2 h_1^T,\ \delta W_3 = -e h_2^T$$

Arild Nøkland
Trondheim, Norway
arild.nokland@gmail.com

Error Propagation Variants



Grey Arrows: Activation Paths

Black Arrows: Error Paths

W: Adapted Weights during Learning

B: Fixed Weights randomly generated

Arild Nøkland
Trondheim, Norway
arild.nokland@gmail.com