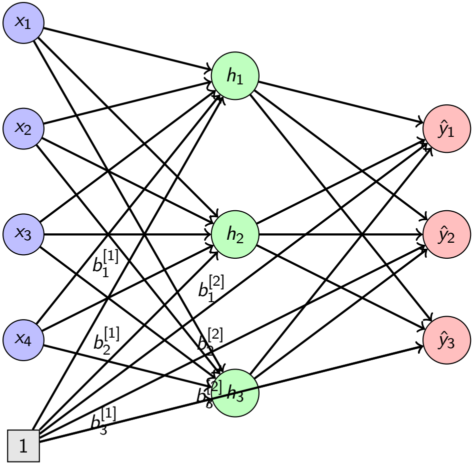
Step-by-Step Neural Network on Iris Dataset

Bindeshwar Singh Kushwaha
Post Network Academy

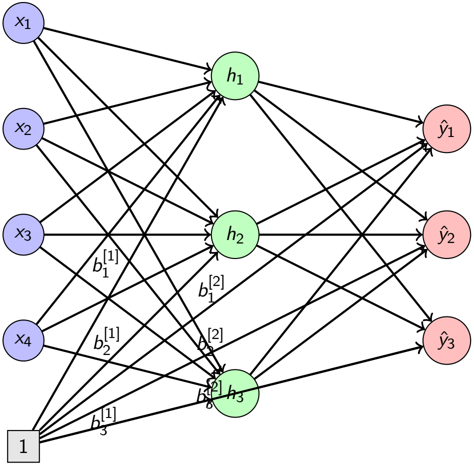
Iris Classification Task

- Dataset: Iris (3 classes, 4 features)
- Example input: $x = [5.1, 3.5, 1.4, 0.2]$
- Target: Setosa $y = [1, 0, 0]$



Network Architecture

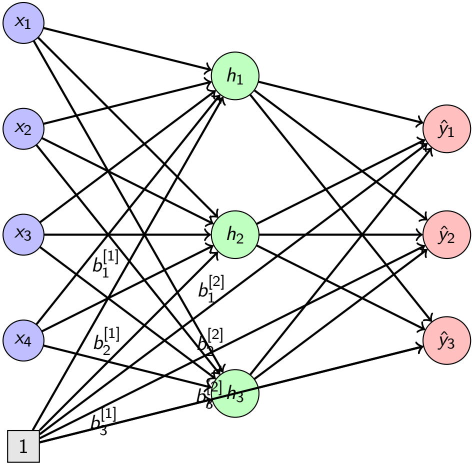
4 inputs $\rightarrow$ 3 hidden $\rightarrow$ 3 outputs



Weights and Biases

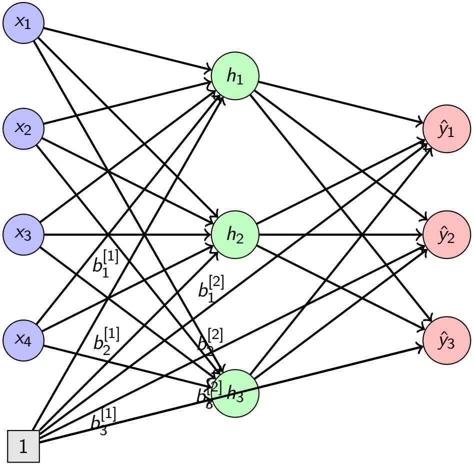
$$W^{[1]} = \begin{bmatrix} 0.1 & 0.2 & 0.1 \\ 0.4 & 0.3 & 0.2 \\ 0.2 & 0.5 & 0.3 \\ 0.1 & 0.2 & 0.4 \end{bmatrix}, \; b^{[1]} = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}$$

$$W^{[2]} = \begin{bmatrix} 0.2 & 0.1 & 0.3 \\ 0.3 & 0.2 & 0.2 \\ 0.4 & 0.2 & 0.1 \end{bmatrix}, \; b^{[2]} = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}$$



Hidden Pre-Activation

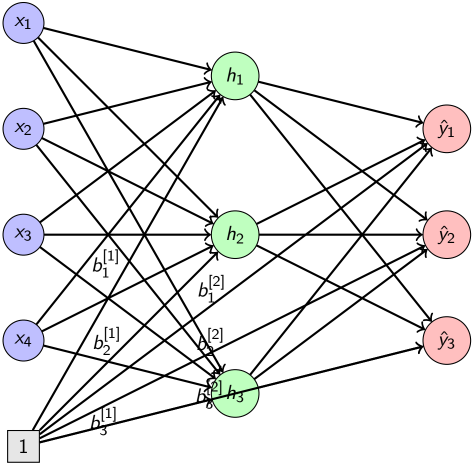
$$z^{[1]} = W^{[1]}x + b^{[1]} = \begin{bmatrix} 2.52 \\ 3.30 \\ 2.60 \end{bmatrix}$$



Hidden Activation

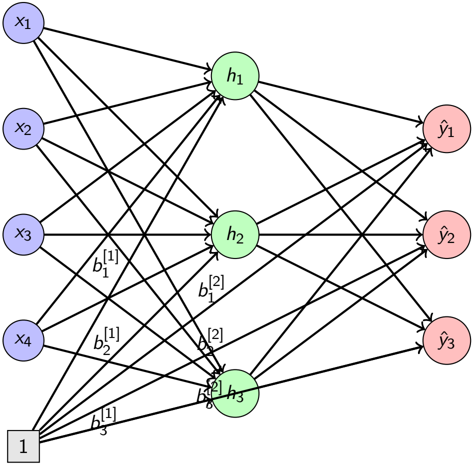
Sigmoid:

$$h = \sigma(z^{[1]}) = \begin{bmatrix} 0.925 \\ 0.964 \\ 0.931 \end{bmatrix}$$



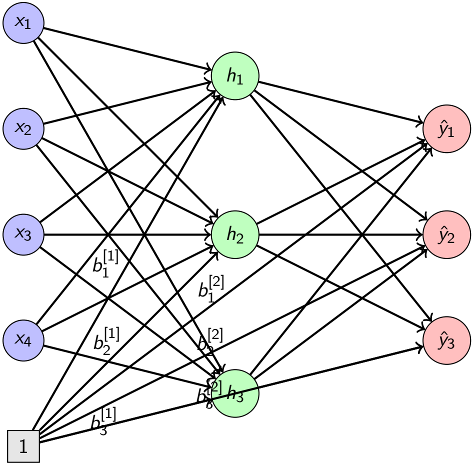
Output Pre-Activation

$$z^{[2]} = W^{[2]}h + b^{[2]} = \begin{bmatrix} 0.971 \\ 0.932 \\ 0.930 \end{bmatrix}$$



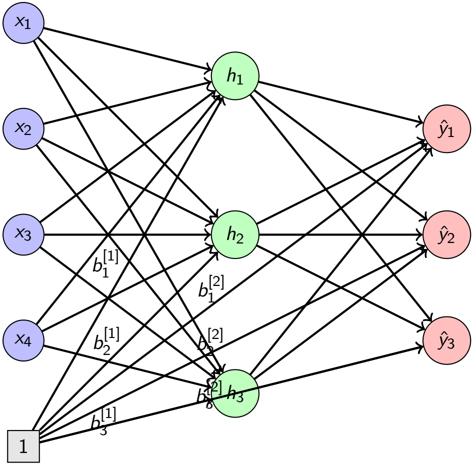
Softmax Probabilities

$$\hat{y} = \text{softmax}(z^{[2]}) = \begin{bmatrix} 0.343 \\ 0.330 \\ 0.327 \end{bmatrix}$$



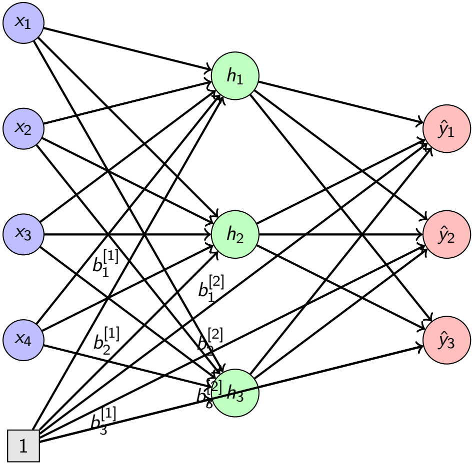
Cross-Entropy Loss

$$L = -\log(\hat{y}_1) = -\log(0.343) \approx 1.07$$



Output Layer Gradient

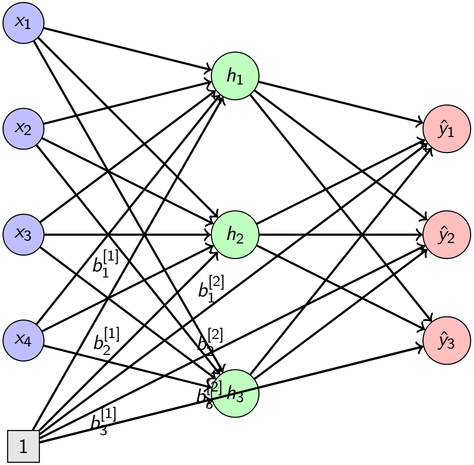
$$\delta^{[2]} = \hat{y} - y = \begin{bmatrix} -0.657 \\ 0.330 \\ 0.327 \end{bmatrix}$$



Hidden Layer Gradient

$$\delta^{[1]} = (W^{[2]})^T \delta^{[2]} \odot \sigma'(z^{[1]})$$

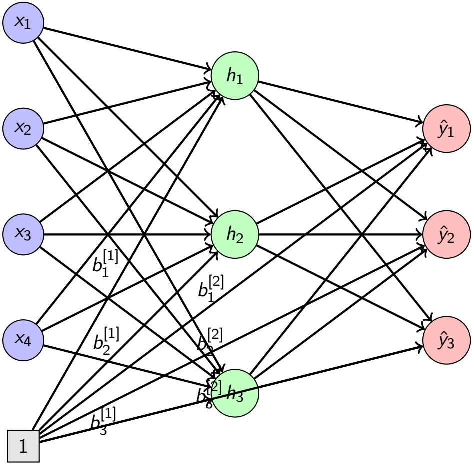
$$\delta^{[1]} \approx \begin{bmatrix} 0.013 \\ 0.011 \\ 0.018 \end{bmatrix}$$



Gradients for $W^{[2]}$

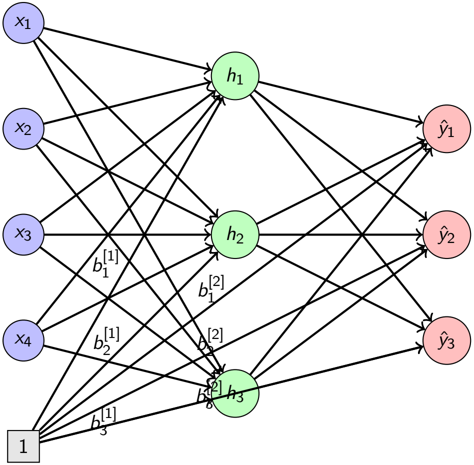
$$\nabla W^{[2]} = \delta^{[2]} h^T$$

$$= \begin{bmatrix} -0.607 & -0.633 & -0.612 \\ 0.305 & 0.318 & 0.307 \\ 0.302 & 0.314 & 0.304 \end{bmatrix}$$



Gradients for $b^{[2]}$

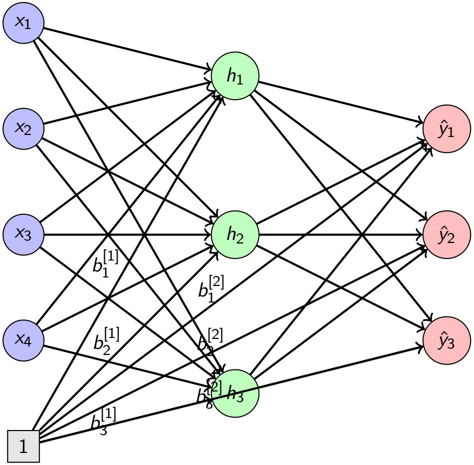
$$\nabla b^{[2]} = \delta^{[2]} = \begin{bmatrix} -0.657 \\ 0.330 \\ 0.327 \end{bmatrix}$$



Gradients for $W^{[1]}$

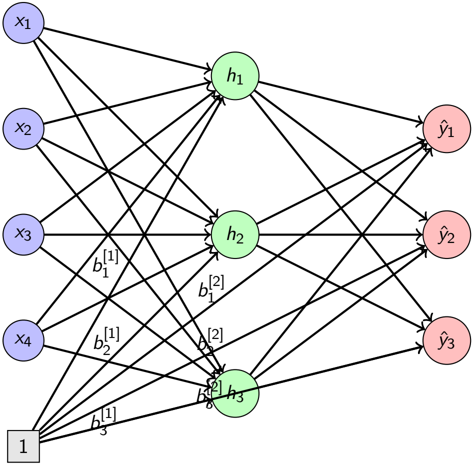
$$\nabla W^{[1]} = \delta^{[1]} x^T$$

$$\approx \begin{bmatrix} 0.066 & 0.046 & 0.018 & 0.003 \\ 0.056 & 0.039 & 0.015 & 0.002 \\ 0.092 & 0.064 & 0.025 & 0.004 \end{bmatrix}$$



Gradients for $b^{[1]}$

$$\nabla b^{[1]} = \delta^{[1]} = \begin{bmatrix} 0.013 \\ 0.011 \\ 0.018 \end{bmatrix}$$

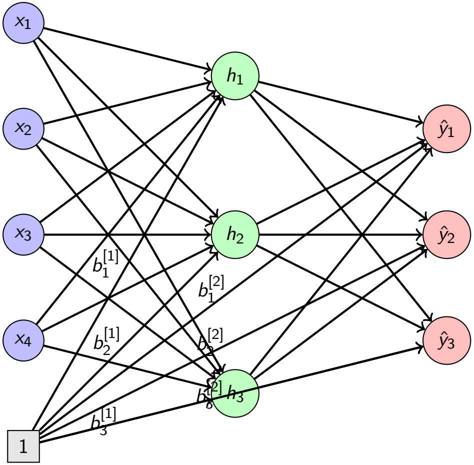


Weight Update Rule

With learning rate $\eta = 0.1$:

$$W^{[l]} \leftarrow W^{[l]} - \eta \nabla W^{[l]}$$

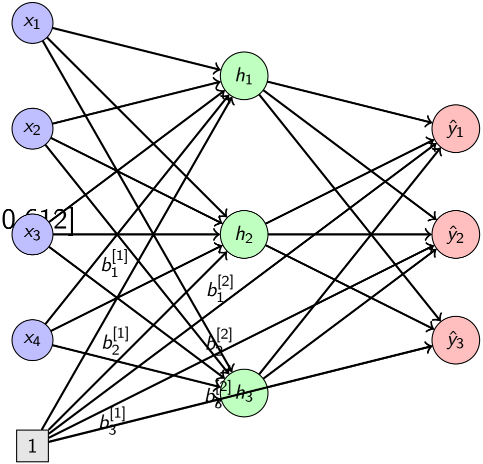
$$b^{[l]} \leftarrow b^{[l]} - \eta \nabla b^{[l]}$$



Updated $W^{[2]}$ (first row)

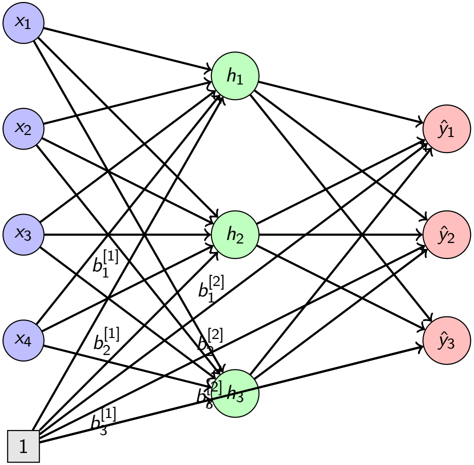
$$W^{[2]}_{\text{new, row 1}} = [0.2, 0.1, 0.3] - 0.1 \times [-0.607, -0.633, -0.612]$$

$$= [0.261, 0.163, 0.361]$$



Updated $b^{[2]}$

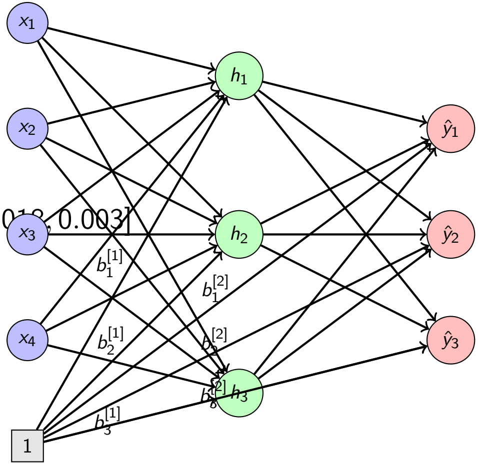
$$b^{[2]}_{\text{new}} = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} - 0.1 \times \begin{bmatrix} -0.657 \\ 0.330 \\ 0.327 \end{bmatrix} = \begin{bmatrix} 0.166 \\ 0.067 \\ 0.067 \end{bmatrix}$$



Updated $W^{[1]}$ (first row)

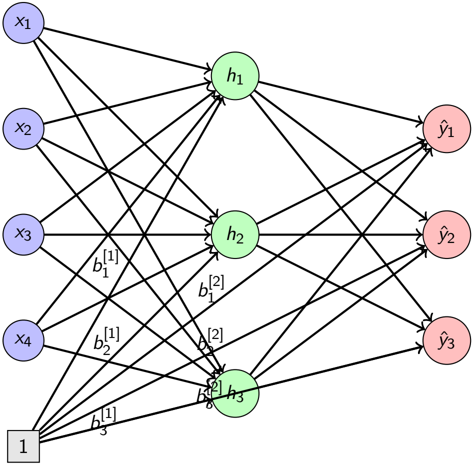
$$W^{[1]}_{\text{new, row 1}} = [0.1, 0.2, 0.1, 0.4] - 0.1 \times [0.066, 0.046, 0.018, 0.003]$$

$$= [0.094, 0.195, 0.098, 0.400]$$



Training Effect

- Loss decreased from 1.07 to lower after one step
- Prediction shifts closer to target
- Iterating improves accuracy



Conclusion

- Forward pass computed numerically
- Backpropagation gave gradients
- Gradient descent updated weights
- Neural network learns Iris classification