If you wish to make a neural network from scratch...

Alex Gezerlis

23rd STFC Nuclear Physics Summer School
University of Aberdeen
August 2026

Neural network

Fancy non-linear least squares fitting

Minimize $$\chi^2 = \frac{1}{N}\sum_{j=0}^{N-1}\left[y_j - p(x_j)\right]^2 \equiv \frac{1}{N}\sum_{j=0}^{N-1} s_j$$

For $$p(x) = \phi\left(\sum_{k=0}^{M-1} a_k\,\phi(b_k x + c_k) + d\right)$$

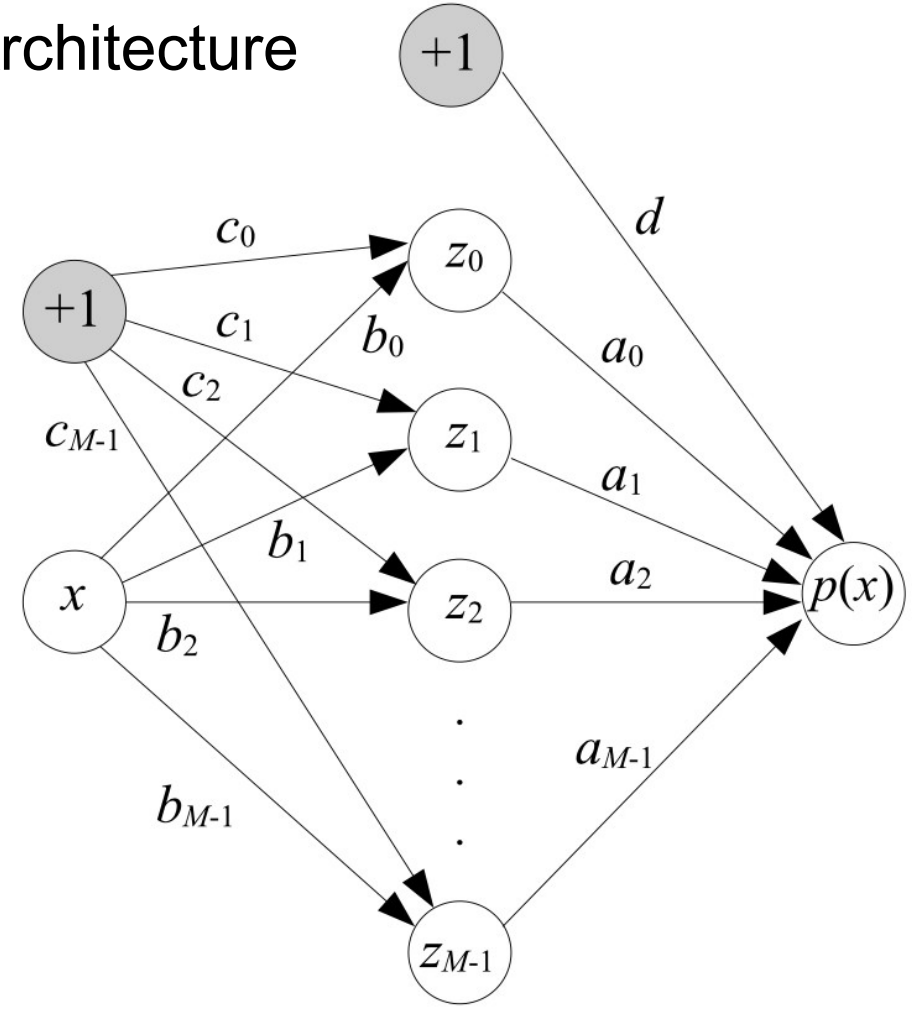
If it helps, consider

$$p(x) = \phi\left(\sum_{k=0}^{M-1} a_k z_k + d\right), \qquad z_k = \phi(b_k x + c_k)$$

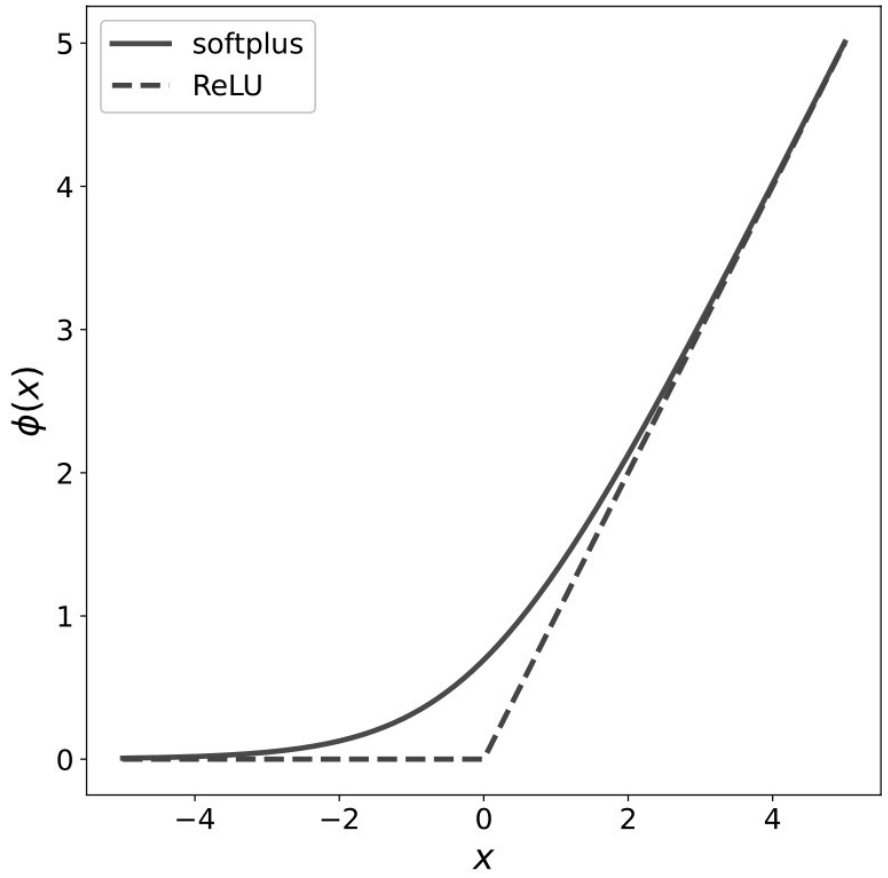
Neural network

$$p(x) = \phi\left(\sum_{k=0}^{M-1} a_k z_k + d\right), \qquad z_k = \phi(b_k x + c_k)$$

Architecture



Activation function $\phi(x) = \ln(1 + e^x)$



Neural network

UNIVERSITY OF GUELPH

Determine your parameters via

Gradient descent: $$\mathbf{w}^{(i)} = \mathbf{w}^{(i-1)} - \gamma \nabla \chi^2(\mathbf{w}^{(i-1)}) = \mathbf{w}^{(i-1)} - \frac{\gamma}{N} \sum_{j=0}^{N-1} \nabla s_j(\mathbf{w}^{(i-1)})$$

Stochastic gradient descent: $$\mathbf{w}^{(i)} = \mathbf{w}^{(i-1)} - \gamma \nabla s_j(\mathbf{w}^{(i-1)})$$

More explicitly: $$a_m^{(i)} = a_m^{(i-1)} - \gamma \left.\frac{\partial s_j}{\partial a_m}\right|^{(i-1)}, \qquad b_m^{(i)} = b_m^{(i-1)} - \gamma \left.\frac{\partial s_j}{\partial b_m}\right|^{(i-1)},$$

$$c_m^{(i)} = c_m^{(i-1)} - \gamma \left.\frac{\partial s_j}{\partial c_m}\right|^{(i-1)}, \qquad d^{(i)} = d^{(i-1)} - \gamma \left.\frac{\partial s_j}{\partial d}\right|^{(i-1)}$$

Neural network

UNIVERSITY OF GUELPH

Backpropagation:

$$\frac{\partial s_j}{\partial a_m} = 2[p(x_j) - y_j]\frac{\partial p(x_j)}{\partial a_m}$$

$$\frac{\partial p(x_j)}{\partial a_m} = \phi'\left(\sum_{k=0}^{M-1} a_k z_k + d\right) z_m$$

$$\frac{\partial s_j}{\partial a_m} = 2[p(x_j) - y_j]\,\phi'\left(\sum_{k=0}^{M-1} a_k z_k + d\right) z_m$$

Similarly:

$$\frac{\partial s_j}{\partial b_m} = 2[p(x_j) - y_j]\,\phi'\left(\sum_{k=0}^{M-1} a_k z_k + d\right) a_m\, x_j\, \phi'(b_m x_j + c_m)$$

$$\frac{\partial s_j}{\partial c_m} = 2[p(x_j) - y_j]\,\phi'\left(\sum_{k=0}^{M-1} a_k z_k + d\right) a_m\, \phi'(b_m x_j + c_m),$$

$$\frac{\partial s_j}{\partial d} = 2[p(x_j) - y_j]\,\phi'\left(\sum_{k=0}^{M-1} a_k z_k + d\right)$$

Neural network

UNIVERSITY OF GUELPH

Your task:

Code this up, following the Unix philosophy
(write functions that do one thing and do it well)

Needed components:

- Generate pseudodata
- Activation function
- Compute z's
- Backpropagation
- Implement stochastic gradient descent

Acknowledgments

UNIVERSITY OF GUELPH

Funding