Week Fourteen PHY-480

Lewis

December 2025

Q27.1

An epoch is one full pass through the entire training set. For each example, the network performs forward propagation to compute the output, evaluates the loss, then uses backward propagation to compute gradients. After all training examples are processed, gradient descent updates the weights and biases once. Thus, an epoch links the training set, forward pass, backward pass, and parameter updates.

Q27.2

In matrix form, a perceptron is written as

$$z = Wx + b, \qquad a = f(z),$$

where W is the weight matrix, b is the bias, and f is the activation.

The *Hadamard product* is element-wise multiplication:

$$(u * v)_k = u_k v_k.$$

The *outer product* of vectors u and v is the matrix

$$u \otimes v = u\, v^T,$$

used to form weight gradients.

Q27.3

Forward propagation for the SK-GNN uses both interlayer weights and intralayer SK couplings. For each layer l:

$$z^{(l)} = \alpha\, W^{(l)} a^{(l-1)} + (1-\alpha)\, J a^{(l)} + b^{(l)}, \qquad a^{(l)} = \tanh(z^{(l)}).$$

The algorithm is:

1. Start with input activations $a^{(1)}$.
2. Compute interlayer term $W^{(l)} a^{(l-1)}$.
3. Compute intralayer term $J a^{(l)}$.

4. Form $z^{(l)}$ using the weighted sum above.
5. Apply tanh to get $a^{(l)}$.

This produces the forward-propagated output $a^{(L)}$.