Deep Learning Formulas and Examples

1. Perceptron Networks

Formulas:

- **Weighted Sum:** z = sum(w_i * x_i + b)
  - w_i: weights
  - x_i: input features
  - b: bias

- **Activation Function (Step Function):**
  output = 1 if z > 0 else 0

Example:

- Inputs: x = [1, 0, 1]
- Weights: w = [0.5, -0.6, 0.2]
- Bias: b = 0.1
- z = 0.5 * 1 + (-0.6) * 0 + 0.2 * 1 + 0.1 = 0.8
- Output: 1 (since z > 0)

2. Backpropagation

Formulas:

- **Loss Function (Mean Squared Error):**
  L = 1/2 sum((y_i - y_hat_i)^2)

- **Gradient of Loss w.r.t Weights:**

  dL/dw_j = sum((y_i - y_hat_i) * y_hat_i * (1 - y_hat_i) * x_ij)

Example:

- Predicted output: y_hat = [0.4, 0.6]

- True output: y = [0.5, 0.5]

- L = 1/2 * [(0.5 - 0.4)^2 + (0.5 - 0.6)^2] = 0.01

3. Convolutional Neural Networks (CNNs)

Formulas:

- **Convolution Operation:**

  (I * K)(i, j) = sum(sum(I(i+m, j+n) * K(m, n)))

- **Pooling Operation (Max Pooling):**

  output(i, j) = max(region)

Example:

- Input Image: I = [[1, 2, 0], [4, 5, 6], [7, 8, 9]]

- Kernel: K = [[1, 0], [-1, 1]]

- Convolution Output: [[2, 3], [5, 6]]

4. Recurrent Neural Networks (RNNs)

Formulas:

- **Hidden State:**

h_t = sigma(W_xh * x_t + W_hh * h_{t-1} + b_h)

- **Output:**

  o_t = sigma(W_ho * h_t + b_o)

Example:

- Input: x_t = [1, 0.5]

- Previous hidden state: h_{t-1} = [0.2, -0.1]

- Weights: W_xh = [[0.1, 0.3], [0.2, -0.1]], W_hh = [[0.5, 0.4], [0.3, 0.2]]

- Bias: b_h = [0.1, -0.1]

- h_t = tanh([[0.1, 0.3], [0.2, -0.1]] * [1, 0.5] + [[0.5, 0.4], [0.3, 0.2]] * [0.2, -0.1] + [0.1, -0.1]) = tanh([0.45, 0.2

5. Long Short-Term Memory (LSTM)

Formulas:

- **Forget Gate:**

  f_t = sigma(W_f * [h_{t-1}, x_t] + b_f)

- **Input Gate:**

  i_t = sigma(W_i * [h_{t-1}, x_t] + b_i)

  tilde_C_t = tanh(W_C * [h_{t-1}, x_t] + b_C)

- **Cell State:**

  C_t = f_t * C_{t-1} + i_t * tilde_C_t

- **Output Gate:**

  o_t = sigma(W_o * [h_{t-1}, x_t] + b_o)

  h_t = o_t * tanh(C_t)

Example:

- Input: x_t = [1, 0.5]

- Previous hidden state: h_{t-1} = [0.2, -0.1]

- Previous cell state: C_{t-1} = [0.5, 0.3]

- Weights: W_f, W_i, W_C, W_o are weight matrices

- Biases: b_f, b_i, b_C, b_o are biases

6. Autoencoders

Formulas:

- **Encoding:**

  h = sigma(W_e * x + b_e)

- **Decoding:**

  x_hat = sigma(W_d * h + b_d)

Example:

- Input: x = [1, 0.5]

- Weights: W_e = [[0.5, 0.4], [0.3, 0.2]], W_d = [[0.5, 0.3], [0.4, 0.2]]

- Biases: b_e = [0.1, -0.1], b_d = [0.2, 0.1]

- h = tanh([[0.5, 0.4], [0.3, 0.2]] * [1, 0.5] + [0.1, -0.1]) = tanh([0.85, 0.4])

7. Variational Autoencoders (VAE)

Formulas:

- **Encoder:**

  mu = f_mu(x)

  log sigma^2 = f_log sigma^2(x)

- **Sampling:**

  z = mu + sigma * epsilon

  epsilon ~ N(0, 1)

- **Decoder:**

  x_hat = g(z)

- **Loss:**

  L = E_q(z|x) [log p(x|z)] - D_KL(q(z|x) || p(z))

Example:

- Input: x = [1, 0.5]

- Encoder networks provide mu and log sigma^2

- Sample z from N(mu, sigma^2)

- Decoder reconstructs x_hat

8. Generative Adversarial Networks (GANs)

Formulas:

- **Generator Loss:**

$$L_G = - E_{z \sim p_z(z)} [\log D(G(z))]$$

- **Discriminator Loss:**

$$L_D = - E_{x \sim p_{data}(x)} [\log D(x)] - E_{z \sim p_z(z)} [\log (1 - D(G(z)))]$$

Example:

- Random noise z is fed into the generator G
- Generated sample G(z) is evaluated by the discriminator D
- Discriminator updates its parameters to differentiate real data from generated data
- Generator updates its parameters to improve the realism of generated samples