

### Soal 1

Berikut proses pelatihan pada ANN dengan :

1. Inisialisasi ukuran dari ANN.
2. Inisialisasi parameter  $w$  dengan nilai kecil random dan  $b$  dengan nilai 0.
3. Pecah data menjadi  $n$  bagian sesuai dengan *batch-size* yang dipilih.
4. Ambil data bagian ke- $i$  untuk dimasukkan ke dalam model.
5. Lakukan proses forward propagation dengan urutan kalkulasi sebagai berikut (superscript (i) merupakan index layer ke-i):

Catatan:  $f(x)$  merupakan suatu fungsi aktivasi. Setiap layer bisa memiliki fungsi aktivasi yang berbeda-beda. Contoh: Sigmoid, ReLU, tanh, Leaky ReLU, dll.

Input Layer

$$Z^{(1)} = X \cdot W^{(1)} + b^{(1)}$$

$$A^{(1)} = f(Z^{(1)})$$

Hidden Layer (layer ke-L)

$$Z^{(L)} = A^{(L-1)} \cdot W^{(L)} + b^{(L)}$$

$$A^{(L)} = f(Z^{(L)})$$

Output Layer

$$Z^{(out)} = A^{(out-1)} \cdot W^{(out)} + b^{(L)}$$

$$A^{(out)} = f(Z^{(out)})$$

6. Hitung loss dari hasil forward propagation (loss function yang digunakan di program ini adalah Mean Squared Error (MSE)).

$$L(W, b) = \frac{1}{m} \sum_{i=1}^m (Y_i - A_i^{(out)})^2$$

7. Lakukan proses backpropagation dengan urutan kalkulasi sebagai berikut:

Catatan: \* adalah operasi element-wise,  $A^{(0)} = X$

Output layer

$$dA^{(out)} = \frac{\partial L(W, b)}{\partial A^{(out)}} = \frac{2}{m} (A^{(out)} - Y)$$

$$dZ^{(out)} = dA^{(out)} * f'(Z^{(out)})$$

$$dW^{(out)} = A^{(out-1)} \cdot dZ^{(out)}$$

$$db^{(out)} = \text{sum}(dZ^{(out)}, \text{axis} = 0) \rightarrow \text{sesuaikan dengan ukuran matrix } b$$

Hidden layer (layer ke-L)

$$dA^{(L)} = dZ^{(L+1)} \cdot W^{(L+1)}$$

$$dZ^{(L)} = dA^{(L)} \cdot f'(Z^{(L)})$$

$$dW^{(L)} = A^{(L-1)} \cdot dZ^{(L)}$$

$$db^{(L)} = \text{sum}(dZ^{(L)}, \text{axis} = 0)$$

Persamaan pada tahap backpropagation didapatkan dengan cara menghitung turunan parsial setiap parameter terhadap loss function menggunakan chain rule.

Ukuran matrix tiap-tiap parameter disesuaikan, yang terpenting adalah untuk suatu parameter tertentu, ukuran matriksnya sama dengan ukuran matriks gradiennya. Jadi ukuran matriks  $W$  harus sama dengan  $dW$ ,  $b$  sama dengan  $db$ , dan yang lainnya.

8. Perbarui parameter  $w$  dan  $b$  dengan cara sebagai berikut:

$$W^{(i)} := W^{(i)} - \alpha * dW^{(i)}$$

$$b^{(i)} := b^{(i)} - \alpha * db^{(i)}$$

9. Ulangi poin 4-8 hingga seluruh  $n$  bagian data sudah masuk ke dalam model.
10. Proses dari 4-9 merupakan 1 epoch. Ulangi proses tersebut sampai mencapai jumlah epoch yang diinginkan.

## Soal 2

Activation function merupakan komponen neural network yang menambahkan properti non-linear kepada model. Properti non-linear sangat berguna untuk berhadapan dengan data yang tidak dapat dipisahkan secara linear, sehingga membuat model lebih fleksibel. Beberapa fungsi lain selain ReLU dan Sigmoid diantaranya adalah fungsi hyperbolic tangent (tanh), Leaky ReLU, dan Parametric ReLU (PReLU).

Loss function merupakan suatu pengukuran seberapa jauh nilai yang diprediksi oleh model dengan nilai sebenarnya. Fungsi inilah yang akan diminimalisasi oleh model melalui proses pelatihan.

Penggunaan berbagai macam loss function disesuaikan dengan jenis permasalahan apa yang sedang dihadapi. Misalkan pada permasalahan regresi, salah satu metric yang tepat untuk digunakan adalah Mean Squared Error (MSE). Untuk permasalahan *binary classification* salah satu metric yang dapat digunakan adalah Binary Cross-Entropy (BCE). Untuk permasalahan *multiclass classification* beberapa metric yang dapat digunakan adalah Categorical Cross-Entropy (CCE) dan Sparse Categorical Cross-Entropy.

Optimizer merupakan suatu algoritma yang digunakan untuk mengubah parameter saat pelatihan sedang berlangsung. Optimizer memiliki fungsi utama untuk meminimalisasi nilai dari loss function. Setiap jenis optimizer memiliki asumsi dan hyperparameter tersendiri, dan hal tersebut dapat berefek ke seberapa cepat model mengerucut, robustness, dan juga kemampuan untuk generalisasi pola. Beberapa contoh optimizer adalah Gradient Descent, Stochastic Gradient Descent, Mini-batch Gradient Descent, RMSprop, Adam (paling populer untuk saat ini), dan sebagainya.

## Soal 3

ANN yang diimplementasikan pada tugas ini adalah ANN jenis Feedforward Neural Network (FNN) yang terdiri dari input layer, satu atau lebih hidden layer, dan juga output layer. Alur propagasi berawal dari input layer menuju output layer, lalu sebaliknya untuk proses backpropagation. Beberapa jenis arsitektur lain selain FNN adalah sebagai berikut:

1. Convolutional Neural Network (CNN):  
CNN memanfaatkan konsep konvolusi untuk menangkap pola pada data secara sekaligus pada daerah tertentu. Biasanya digunakan pada data visual seperti gambar.
2. Recurrent Neural Network (RNN):  
RNN didesain supaya informasi dari layer-layer sebelumnya dapat berpengaruh pada layer yang ada didepannya (tidak harus persis di depannya, tapi bisa juga lompat beberapa layer). RNN didesain seperti itu supaya bisa memanfaatkan urutan data sebagai salah faktor dalam mengeluarkan suatu output. Biasanya digunakan pada data sequential seperti teks (NLP), time-series, dan juga speech recognition.
3. Generative Adversarial Network (GAN):  
Terdapat 2 network utama pada GAN, yaitu generator dan discriminator. Generator bertujuan untuk membangkitkan suatu output berdasarkan data yang sudah ada, sementara discriminator bertujuan untuk membedakan antara data asli dan data hasil pembangkitan. Dua network ini dilatih secara adversarial (berlawanan satu sama lain). Biasanya digunakan untuk augmentasi data dan pembangkitan gambar.
4. Autoencoder (AE):  
Jenis neural network yang bersifat unsupervised. AE mempunyai encoder untuk meng-compress data ke dimensi yang lebih kecil kemudian di-restore kembali ke dimensi semula menggunakan decoder.