Bagian B: Implementasi Mini-batch Gradient Descent
Dikumpulkan: Sabtu, 26 Maret 2022 jam 23.59 waktu Edunex.

1. Implementasi backpropagation dengan mini-batch gradient descent sesuai materi kuliah.
    a. Pada mini-batch gradient descent, update bobot dilakukan per mini-batch. Satu train data dibagi menjadi beberapa mini-batch sesuai parameter **batch_size**.
    b. Setiap layer dibatasi memiliki neuron dengan fungsi aktivasi yang sama, sedangkan antar layer diperbolehkan memiliki neuron dengan fungsi aktivasi yang berbeda.
    c. Algoritma yang diimplementasikan adalah backpropagation dengan fungsi **aktivasi** linear, sigmoid, ReLU, dan softmax.
    d. Untuk linear, sigmoid, dan ReLU, gunakan fungsi loss berupa sum of squared errors: $E = \frac{1}{2} \sum_{k \in outputs} (t_k - o_k)^2$

       Untuk softmax, gunakan fungsi loss berupa cross entropy:

       $$E = -\log(p_k), \text{k=target}$$

    e. Turunan dari fungsi aktivasi:

       Linear: f(x)=x → df/dx=1.

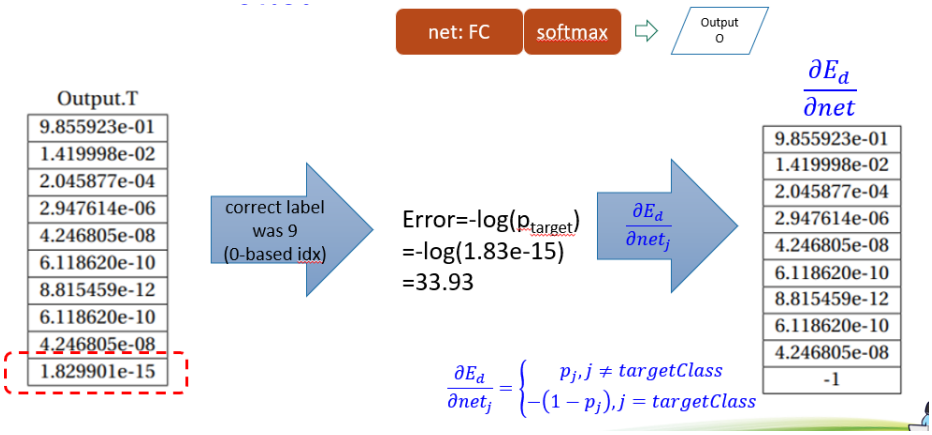
       $$\frac{d}{dx} relu(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

       $$\frac{d}{dx} sigmoid(x) = sigmoid(x)(1 - sigmoid(x))$$

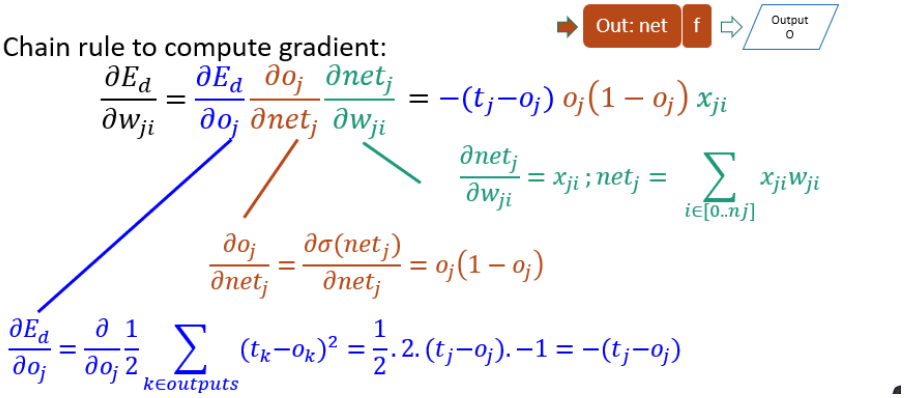
       Softmax:

       $$\frac{\partial E_d}{\partial net_j} = \begin{cases} p_j, j \neq targetClass \\ -(1 - p_j), j = targetClass \end{cases}$$

       Contoh perhitungan turunan softmax:



f. Dalam mengimplementasikan backpropagation, implementasi berupa perhitungan gradient dengan hasil aturan rantai sebelum mengupdate bobot dengan cara mengalikan gradient*-1***learning rate**. Perhatikan aturan rantai perhitungan gradient utk update bobot ke output layer, berbeda dengan hidden layer. Untuk update bobot output layer w yang menerima input h, gradient dihitung sebagai dE/dw=dE/net*dnet/dw. Untuk aktivasi output neuron selain softmax, diimplementasikan dE/dw=dE/dOut*dOut/dNet*dNet/dw sehingga kita mengalikan hasil dari 3 suku yaitu -(t-out), out(1-out), dan x. Pada softmax, sudah diberikan langsung dE/net sehingga kita mengalikan dE/dw=dE/dNet*dNet/dw.



g. Kondisi berhentinya adalah error kumulatif <=**error threshold** atau **maksimum iterasi** tercapai. Error threshold dan maksimum iterasi merupakan parameter dari mini-batch gradient descent.

h. Definisikanlah parameter apa saja yang bisa ditangani dalam implementasi kelompok Anda. Parameter yang wajib ada adalah: struktur jaringan (jumlah layer, jumlah neuron setiap layer, fungsi aktivasi setiap layer), learning-rate, error threshold, max_iter, batch_size.

i. Pengujian kebenaran fungsional dari backprop yang diimplementasikan dilakukan dengan menggunakan kasus uji yang disiapkan oleh asisten (TBD)

    j. Lakukan pembelajaran backpropagation dengan hasil implementasi ini untuk dataset iris untuk semua data (full training), dan menampilkan modelnya.
2. Deliverables: a) source code, b) laporan berisi penjelasan implementasi, hasil eksekusi (langkah 2), perbandingan dengan hasil MLP sklearn, dan pembagian tugas setiap anggota kelompok.