**Machine Learning**

**Phạm Gia Phong - 21020128**

Question 1: Perceptron

Bài tập này chúng ta sẽ phân loại nhị phân dựa trên Perceptron và hoàn thành trong class *PerceptronModel*.

Chúng ta sẽ làm việc trên 3 hàm là *run(), get\_prediction()* và *train()*

* *run()*: Trả về phép nhân ma trận giữa input x và trọng số w, sử dụng hàm *nn.DotProduct()*
* *get\_prediction()*: Dự đoán: trả về 1 nếu kết quả hàm *run* ở trên ≥ 0 và trả về -1 nếu ngược lại. Sử dụng hàm *nn.as\_scalar(run())* để chuyển dữ liệu dạng Node sang Python floating-point number (số thực)
* *train()*: sử dụng hàm *update()* trong class *nn.Parameter* và vòng lặp While chạy qua toàn bộ dataset, khi dự đoán x ≠ y, chúng ta sẽ update lại trọng số



Question 2: Non-linear Regression

Bài tập này chúng ta sẽ xây dựng model Regression với hàm kích hoạt Relu() để thành non-linear Regression ở class *RegressionModel*

Chúng ta sẽ làm việc trên 4 hàm là *\_init()\_, run(), get\_loss()* và *train()*

* *\_init()\_*: Khởi tạo learning\_rate alpha, khởi tạo batch\_size, khởi tạo model với 3 layers (2 hidden layers) bằng cách sử dụng hàm *nn.Parameter()*
* *run()*: Xây dựng từng layer bằng cách nhân input của layer này với trọng số w và cộng thêm bias, thêm hàm Relu() cho 2 hidden layers, kết quả sẽ là đầu vào của layer tiếp theo.
* *get\_loss()*: Trả về kết quả là trung bình của bình phương hiệu giữa dự đoán và kết quả chia cho 2
* *train()*: dùng vòng lặp while và hàm update các parameter w và b với thuật toán gradient cho đến khi kết quả của loss ≤ 0.005 thì dừng lại

Question 3: Digit Classification

Bài tập này chúng ta sẽ xây dựng model nhận dạng chữ viết tay từ MNIST dataset ở class *DigitClassificationModel*

Chúng ta làm việc với 4 hàm *\_init()\_, run(), get\_loss()* và *train()*

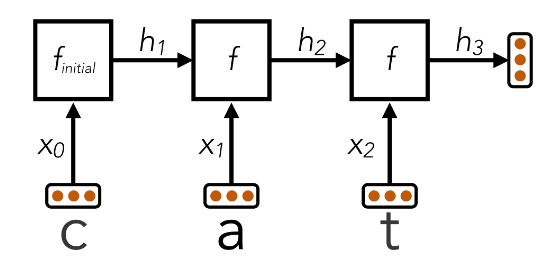
* *\_init()\_*: Khởi tạo learning\_rate alpha, khởi tạo batch\_size, khởi tạo model với 4 layers (3 hidden layers) bằng cách sử dụng hàm *nn.Parameter()*
* *run()*: Xây dựng từng layer bằng cách nhân input của layer này với trọng số w và cộng thêm bias, thêm hàm Relu() cho 3 hidden layers, kết quả sẽ là đầu vào của layer tiếp theo.
* *get\_loss()*: Trả về kết quả hàm loss cho softmax dựa vào xác suất của 10 label (sử dụng batched softmax loss để cập nhật trọng số)
* *train()*: dùng vòng lặp while và hàm update các parameter w và b với thuật toán gradient cho đến khi kết quả của accuracy ≥ 0.97 thì dừng lại

Question 4: Language Identification

Bài tập này chúng ta sẽ xây dựng model nhận dạng 5 ngôn ngữ ở class *LanguageIDModel*

Chúng ta làm việc với 4 hàm *\_init()\_, run(), get\_loss()* và *train()*

* *\_init()\_*: Khởi tạo số chiều ẩn của từ, learning\_rate alpha, batch\_size, model với độ sâu như hình và các trọng số w và b



* *run()*: Ở layer đầu tiên thì chỉ có input nhân với trọng số input, từ các layer thứ 2 đến trước layer cuối cùng đầu ra sẽ là phép cộng 2 ma trận input và ma trận trọng số của layer trước đó. Layer cuối là kết quả của hàm kích hoạt ReLu() đầu ra của layer liền trước
* *get\_loss()*: Trả về kết quả hàm loss cho softmax dựa vào xác suất của 5 label (sử dụng batched softmax loss để cập nhật trọng số)
* *train()*: dùng vòng lặp while và hàm update các parameter w và b với thuật toán gradient cho đến khi kết quả của accuracy ≥ 0.89 thì dừng lại