MAC0460 – EP4

DCC / IME-USP — Primeiro semestre de 2019

Entrega até: 13/05 — Este EP deve ser feito individualmente

Objetivos: O objetivo do EP4 é ampliar a familiarização com (1) os algoritmos Perceptron Learning Algorithm (PLA), Perceptron pocket, regressão linear e regressão logística, por meio da aplicação dos mesmos no problema de classificação de dígitos manuscritos; (2) a transformação da representação: neste EP, de imagens (*pixels*) para um par de *features*; (3) alguns conceitos e ferramentas úteis no processo de treinamento de algoritmos de *machine learning*: embaralhar dados, plotar superfícies de decisão, plotar curva de evolução da função de custo, plotar matriz de confusão; e (4) o treinamento / teste: uso de um conjunto de teste/validação para estimar E_{out}; diferentes métricas de avaliação de desempenho.

Notebook: Este enunciado é acompanhado de um notebook Python, disponível em https://github.com/MLIME/MAC0460/blob/master/notebooks/EP4_MNIST.ipynb

Usaremos o famoso dataset MNIST que contém imagens (já tratadas) de dígitos manuscritos. Trabalharemos neste EP com classificação binária, especificamente com imagens restritas a duas classes de dígitos (uma das classes será a classe negativa -1 e a outra será a positiva $+1$).

O notebook está preparado para realizar a leitura do MNIST, converter cada imagem de dígito em um par de características (x_1, x_2), sendo x_1 a simetria e x_2 a intensidade média, preparar os dados a serem usados para treinamento e teste.

Para cada algoritmo, também já estão prontos a parte que:

- mostra a evolução da função custo (E_{in}) a partir do histórico retornado pelos algoritmos de treinamento[1] ,
- realiza a predição usando o peso calculado no treinamento, sobre os exemplos do próprio conjunto de treinamento
- plota a superfície de decisão (a reta separadora) e a matriz de confusão, referentes ao conjunto de treinamento.

Tarefa

- Preencher os espaços reservados para cada um dos quatro algoritmos: Perceptron Learning Algorithm (PLA), Perceptron pocket, regressão linear e regressão logística

  Desses quatro, todos exceto o pocket já foram implementados nos EPs anteriores. Vocês podem reaproveitar diretamente a versão já implementada, fazendo os ajustes necessários.

[1] Exceto para a regressão linear pois este tem uma solução de um passo só.

Por exemplo, o PLA deverá ser alterado para devolver o histórico contendo o número de erros e classificação a cada iteração do algoritmo. O pocket perceptron funciona da mesma forma que o perceptron, exceto pelo fato de que ele armazena a melhor solução encontrada até a iteração atual; o histórico deve ser montado com respeito a essa melhor solução. Ao final, deve ser devolvido o vetor peso correspondente à primeira melhor solução geral encontrada durante o treinamento.

Estes algoritmos precisam ser implementados para que o notebook possa ser executado inteiramente.

- Ao final do notebook, acrescentar trecho que explora os diferentes parâmetros no treinamento do algoritmo de **regressão logística**, conforme descrito a seguir.

  1. Variar a quantidade de exemplos N usados no treinamento: N deve ser variado de 1000 a 12000[2], de 1000 em 1000. Para usar uma certa quantidade N de exemplos no treinamento, use os primeiros N exemplos no array X e Y (i.e., `X[:N,:]` e `Y[:N]`). Os demais parâmetros ficam fixos em seu valor default.

     Para este caso, plotar um gráfico com N no eixo x e as seguintes métricas, calculadas sobre o conjunto de teste, no eixo y:

     - E_{test} (cross-entropy loss)
     - acurácia (proporção de acertos de classificação)
     - precision e recall

  2. Fixar N (=todos os exemplos de treinamento) e variar o *batch size*.

     Para este caso, plotar um gráfico com o *batch size* no eixo x e E_{test} e acurácia, calculados sobre o conjunto de teste, no eixo y

  3. Fixar N (=todos os exemplos de treinamento) e variar o *learning rate*.

     Para este caso, plotar um gráfico com o *learning rate* no eixo x e E_{test} e acurácia, calculados sobre o conjunto de teste, no eixo y

  Para cada experimento, adicione blocos de texto ao notebook com descrições claras e comentários relacionados. Deixe claro nestas descrições o que está acontecendo no experimento, e quais conclusões obtidas dos resultados.

O que entregar:

- o html do seu notebook, contendo o resultado da execução completa. Para gerar o html, use o menu `File --> Download as` do Jupyter
- o notebook. O notebook só será executado/considerado caso as informações presentes no html forem ambíguas ou inconclusivas.

Dúvidas? Postem as discussões no "Fórum de discussão" do PACA.

[2] Se eventualmente não existirem 12000 exemplos, este limite pode ser alterado para o maior múltiplo de 1000