IA353 – Redes Neurais (1s2024)
Projeto Computacional – Parte 1 – PCs #01 a #05
Atividade em Dupla – Peso 5
Data de entrega dos resultados solicitados: 16/04/2024

Síntese de classificadores regularizados lineares, lineares generalizados e não-lineares

1 Regressão de quadrados mínimos

- Considere que você tenha à disposição um conjunto de *N* amostras de treinamento na forma: $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, onde $\mathbf{x}_i \in \Re^n$, *i*=1,...,*N*. Suponha também que $N > n$.
- A regressão de quadrados mínimos busca um vetor $\mathbf{w} \in \Re^n$ que minimiza:

$$J(\mathbf{w}) = \sum_{i=1}^{N} \left(\mathbf{x}_i^T \mathbf{w} - y_i \right)^2 .$$

- Fazendo $X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ 1 & x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Nn} \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$ e acrescentando um elemento de *offset* ao vetor **w**, constata-se que a regressão de quadrados mínimos requer a solução de um sistema linear sobredeterminado, na forma:

$$\min_{\mathbf{w}} \left\| X\mathbf{w} - \mathbf{y} \right\|_2^2 .$$

- Caso a matriz *X* tenha posto completo, a Seção 2 fornecerá a solução para este problema de otimização, na forma:

$$\mathbf{w} = \left(X^T X \right)^{-1} X^T \mathbf{y} .$$

- Caso a matriz *X* não tenha posto completo, a Seção 3 fornecerá a solução para este problema de otimização, na forma:

$$\mathbf{w} = \left(X^T X + \lambda I \right)^{-1} X^T \mathbf{y} \text{, com } \lambda > 0 .$$

2 Resolvendo sistemas lineares sobredeterminados

- O sistema $X\mathbf{w} = \mathbf{y}$, com $X \in \Re^{N \times (n+1)}$, $\mathbf{w} \in \Re^{(n+1) \times 1}$, $\mathbf{y} \in \Re^{N \times 1}$ e $N \geq (n+1)$, sendo *X* uma matriz de posto completo, tem garantia de solução exata apenas quando $N = (n+1)$. Na situação em que $N > (n+1)$, há mais equações do que incógnitas, criando a possibilidade de inconsistência entre algumas equações (regidas pelas linhas da matriz *X*), que não podem ser satisfeitas simultaneamente.
- A presença de inconsistência impede, portanto, que exista **w** tal que $X\mathbf{w} = \mathbf{y}$, mas não impede que se busque encontrar **w** que resolva o seguinte problema de programação quadrática:

$$\min_{\mathbf{w}} \sum_{i=1}^{N} \left(\mathbf{x}_i^T \mathbf{w} - y_i \right)^2 = \min_{\mathbf{w}} \left\| X\mathbf{w} - \mathbf{y} \right\|_2^2 = \min_{\mathbf{w}} \left(X\mathbf{w} - \mathbf{y} \right)^T \left(X\mathbf{w} - \mathbf{y} \right)$$

- A função-objetivo fica:

$$J(\mathbf{w}) = \left(X\mathbf{w} - \mathbf{y} \right)^T \left(X\mathbf{w} - \mathbf{y} \right) = \mathbf{w}^T X^T X\mathbf{w} - \mathbf{w}^T X^T \mathbf{y} - \mathbf{y}^T X\mathbf{w} + \mathbf{y}^T \mathbf{y} .$$

- Aplicando a condição necessária de otimalidade, que afirma que o gradiente se anula nos pontos extremos da função-objetivo, resulta:

$$\frac{dJ(\mathbf{w})}{d\mathbf{w}} = 2X^T X\mathbf{w} - 2X^T \mathbf{y} = 0 \Rightarrow X^T X\mathbf{w} = X^T \mathbf{y} .$$

- Se a matriz *X* for de posto completo, então $X^T X$ tem inversa, o que produz:

$$\mathbf{w} = \left(X^T X \right)^{-1} X^T \mathbf{y} .$$

- Esta equação representa a famosa solução de quadrados mínimos para um sistema linear de equações que não necessariamente admite solução exata. Isto implica que $X\mathbf{w} \neq \mathbf{y}$, mas $X\mathbf{w}$ é o mais próximo que se pode chegar de $\mathbf{y}$, no sentido de quadrados mínimos.

3 Quadrados mínimos regularizados

- Tomando o mesmo cenário das Seções 1 e 2, é possível adicionar um termo de regularização, que penaliza o crescimento da norma do vetor **w**, produzindo o problema regularizado de regressão de quadrados mínimos, também denominado de *ridge regression*, na forma:

$$\min_{\mathbf{w}} \left\| X\mathbf{w} - \mathbf{y} \right\|_2^2 + \lambda \left\| \mathbf{w} \right\|_2^2, \ \lambda \geq 0 .$$

- Aplicando a condição necessária de otimalidade, como feito na Seção 2, obtém-se como solução ótima:

$$\mathbf{w} = \left(X^T X + \lambda I \right)^{-1} X^T \mathbf{y} .$$

- A definição de um valor para o parâmetro de regularização $\lambda \geq 0$ pode ser feita por técnicas de validação, conforme será implementado neste projeto computacional. Lembre-se que, quando a matriz *X* não tem posto completo, necessariamente deve-se tomar $\lambda > 0$.
- O nome da técnica, *ridge regression*, está associado ao fato de que a solução passa a incluir um termo que é uma matriz cumeeira, em analogia à figura a seguir.

- O modelo regularizado de regressão linear assume então a forma:

$$\mathbf{w}^T \mathbf{x} = \mathbf{y}^T X \left(X^T X + \lambda I \right)^{-1} \mathbf{x} .$$

- Cabe enfatizar que o peso de bias não deveria sofrer penalização, mas vamos manter essa penalização por simplicidade da formulação.

4 Emprego de múltiplos coeficientes de regularização

- Note que é possível, e deve ser feito na parte experimental, substituir o vetor **y** por uma matriz com tantas colunas quanto classes. Com isso, o vetor **w** também vai corresponder a uma matriz, com o mesmo número de colunas de **y**. A desvantagem desta estratégia é que se força o emprego de um mesmo coeficiente de regularização para todos os classificadores.
- A formulação de quadrados mínimos regularizados vale para uma única saída, conforme segue:

$$\min_{\mathbf{w}} \|X\mathbf{w}-\mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_2^2,\ \lambda \geq 0\,.$$

  onde $X \in \Re^{N\times(n+1)}$, $\mathbf{w} \in \Re^{(n+1)\times 1}$, $\mathbf{y} \in \Re^{N\times 1}$ e $\lambda \in \Re^{1\times 1}$, sendo *N* o número de dados de entrada-saída para treinamento supervisionado.
- Considerando um mesmo índice de regularização para todas as saídas e como **w** e **y** aparecem sem nenhuma multiplicação à direita na solução ótima:

$$\mathbf{w} = \left(X^T X + \lambda I\right)^{-1} X^T \mathbf{y}\,,$$

  então, havendo *r* saídas, os *r* vetores **w** podem ser obtidos simultaneamente, organizando as saídas desejadas de cada classificador como colunas da agora matriz $\mathbf{y} \in \Re^{N\times r}$, produzindo uma solução ótima $\mathbf{w} \in \Re^{(n+1)\times r}$.
- Em termos conceituais, se está resolvendo o seguinte problema de otimização:

$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_r} \sum_{c=1}^{r} \|X\mathbf{w}_c - \mathbf{y}_c\|_2^2 + \lambda \sum_{c=1}^{r} \|\mathbf{w}_c\|_2^2,\ \lambda \geq 0\,,$$

  que tem como soluções ótimas:

$$\mathbf{w}_c = \left(X^T X + \lambda I\right)^{-1} X^T \mathbf{y}_c, c = 1,\ldots,r\,.$$

- No entanto, seria possível tomar um coeficiente de regularização para cada uma das *r* saídas, conduzindo ao problema:

$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_r} \sum_{c=1}^{r} \|X\mathbf{w}_c - \mathbf{y}_c\|_2^2 + \sum_{c=1}^{r} \lambda_c \|\mathbf{w}_c\|_2^2,\ \lambda_c \geq 0, c = 1,\ldots,r\,,$$

  cujas soluções ótimas individuais são como segue:

$$\mathbf{w}_c = \left(X^T X + \lambda_c I\right)^{-1} X^T \mathbf{y}_c, c = 1,\ldots,r\,.$$

  Cabe enfatizar que, para cada dado de entrada a ser classificado, a classe escolhida é aquela de maior saída entre as *r* saídas, o que cria uma dependência entre as saídas que não (necessariamente) é contemplada pelas soluções ótimas individuais.

5 Casos de estudo

Caso de estudo 1

Base de dados MNIST: contém dígitos manuscritos rotulados em 10 classes (são os dígitos de '0' a '9'), sendo 60.000 amostras para treinamento e 10.000 amostras para teste (os dados de teste não devem ser empregados em nenhuma fase do processo de síntese do classificador). Cada imagem de entrada contém 784 pixels (no intervalo [0,255], correspondente a níveis de cinza), visto que a dimensão é 28 × 28 pixels. Considere que a classe 10 corresponde ao dígito '0'.

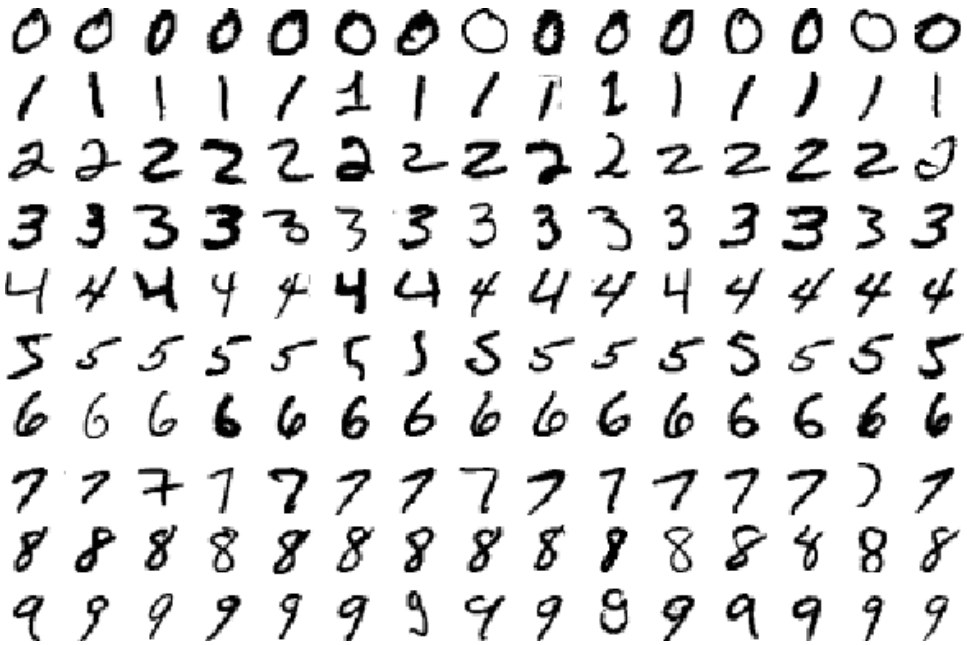

Figura 1 – Exemplos de imagens do conjunto de dados MNIST. Nos dados, a classe '0' é a última, e não a primeira, como na figura acima.

Caso de estudo 2

Base de dados CIFAR-10: é uma coleção de imagens coloridas do *Canadian Institute For Advanced Research*, que são comumente usadas para treinar algoritmos de aprendizado de máquina e visão computacional. Contém 60.000 imagens coloridas 32x32 em 10 classes diferentes. As 10 classes diferentes representam aviões, carros, pássaros, gatos, veados, cães, sapos, cavalos, barcos e caminhões. Há 6.000 imagens de cada classe.

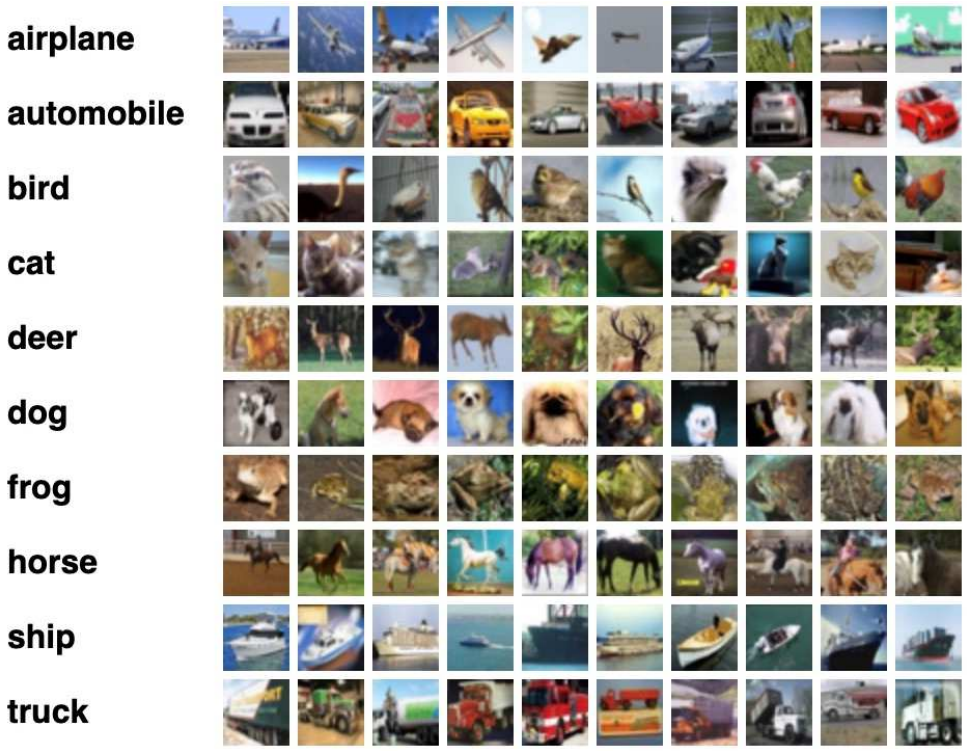

Figura 2 – Exemplos de imagens do conjunto de dados CIFAR-10.

6 Atividades práticas

6.1 *Projeto Computacional 01*

Os dados da base MNIST já foram pré-processados e foi introduzida uma entrada constante como primeira coluna de *X*, de modo que a matriz *X* de dados de treinamento tem dimensão 60.000 por 785. Executando o notebook [**PC01_LC_MNIST.ipynb**], você vai obter um modelo linear de classificação para as 10 classes existentes, de tal modo que a saída para cada classe seja produzida como segue, contendo uma entrada fixa de polarização (*offset*) como primeira coluna da matriz *X*:

$$c_j = w_{0j} + w_{1j}x_1 + w_{2j}x_2 + \cdots + w_{784\,j}x_{784},\ j \in \{1,\cdots,10\},$$

sendo que os parâmetros do modelo linear devem compor uma matriz *W* de dimensão 785 × 10, sendo obtidos de forma fechada, a partir de uma única expressão algébrica. Com isso, o coeficiente de regularização vai ser único para as 10 classes. Deve-se buscar um bom coeficiente de regularização (maior do que zero se a matriz de dados de entrada *X* não tiver posto completo). Para tanto, tomar parte dos dados de treinamento como validação (Sugestão: 70% dos dados para treinamento e 30% para validação), adotando a técnica *holdout*, para poder implementar esta busca pelo melhor coeficiente de regularização, considerando como critério de desempenho para o classificador a taxa de acerto na classificação. O coeficiente de regularização deve ser buscado iniciando pelo seguinte conjunto de valores candidatos:

$$\left\{2^{-10}, 2^{-8}, \cdots, 2^{0}, 2^{+2}, \cdots, 2^{+18}\right\}.$$

Este intervalo de busca é amplo, mas pode não ser adequado para todas as partições a serem produzidas. Uma busca refinada é realizada automaticamente, após concluída a busca grossa. Ela faz uma busca linear no intervalo cujos extremos são os coeficientes imediatamente anterior e posterior àquele sugerido pela busca grossa. Uma vez encontrado um valor adequado para o coeficiente de regularização (após terminadas as buscas grossa e refinada), o programa usa todos os dados de treinamento para sintetizar o classificador linear.

Responda as 5 questões do notebook, que estão em verde e indexadas do item (a) ao item (e). Forneça as respostas diretamente no próprio notebook, nas células indicadas.

6.2 *Projeto Computacional 02*

Executando o notebook [**PC02_LC_CV.ipynb**], você vai obter novamente classificadores lineares para a base MNIST. Em seguida, responda as questões levantadas nos itens (a) e (b) do notebook, que estão em verde. Prossiga executando o mesmo notebook, agora adaptado para a base CIFAR-10. Em seguida, responda as questões levantadas nos itens (c) e (d) do notebook, que estão em verde.

6.3 *Projeto Computacional 03*

O mesmo fluxo de informação adotado no PC01 (classificador linear) pode ser empregado ao se considerar uma máquina de aprendizado extremo (ELM). Execute o notebook [**PC03_ELM.ipynb**] fornecido pelo professor, mas agora para as duas bases: tome 1000 neurônios na única camada intermediária para a base MNIST e

2000 para a base CIFAR-10. Em seguida, responda as questões levantadas nos itens (a) a (d) do notebook, que estão em verde.

6.4 *Projeto Computacional 04*

Tomando os mesmos problemas de classificação de dados das bases MNIST e CIFAR-10, use o framework Keras, tendo o TensorFlow como *backend* e realize o treinamento de uma rede neural MLP. O notebook [**PC04_MLP_PartA.ipynb**] apresenta uma sugestão de código e de configuração de hiperparâmetros. Embora você possa buscar inspiração em resultados já publicados na literatura e/ou adotar um procedimento de tentativa-e-erro para definir, da melhor forma que você puder, o número de camadas intermediárias, o número de neurônios por camada, o algoritmo de ajuste de pesos, a taxa de *dropout* (onde for pertinente) e o número de épocas de treinamento, **execute os notebooks** [**PC04_MLP_MNIST_PartB.ipynb**] e [**PC04_MLP_CIFAR10_PartB.ipynb**], que realizam uma busca em grade simplificada, e **faça uma escolha adequada de hiperparâmetros para a última célula desses dois notebooks**, após conferir os boxplots gerados. Repare que está sendo considerada a média de várias execuções (junto a cada configuração candidata) para se chegar a um índice de desempenho mais estável.

6.5 *Projeto Computacional 05*

Tomando os mesmos problemas de classificação de dados das bases MNIST e CIFAR10 e novamente usando o framework Keras, tendo o TensorFlow como *backend*, realize o treinamento de uma rede neural com camadas convolucionais, usando maxpooling e dropout. O notebook [**PC05_CNN.ipynb**] apresenta uma sugestão de código e de configuração de hiperparâmetros que pode ser adotada. Compare os resultados (em termos de taxa de acerto na classificação) com aqueles obtidos pelos três tipos de máquinas de aprendizado adotadas nas atividades anteriores (classificador linear – escolha um deles –, ELM e MLP), consolidando os resultados numa única tabela, para cada base de dados. Uma busca em grade análoga àquela realizada na Parte B do **PC04** pode ser implementada aqui, mas fica como um item opcional.