FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Técnicas e Algoritmos em Ciência de Dados
Tarefa 3

Este trabalho deve ser enviado até o dia 22 de maio de 2023, às 07h.
Envios atrasados serão penalizados em 10% por hora de atraso.

Tópicos avaliados

Esta tarefa oferece uma oportunidade emocionante para os alunos colocarem em prática seus conhecimentos adquiridos em sala de aula, usando redes neurais para resolver problemas do mundo real, tanto na classificação quanto na regressão. Os alunos aplicarão os conceitos que aprenderam para construir, treinar e otimizar redes neurais, usando um conjunto de validação para ajustar hiperparâmetros. Os alunos também se acostumarão a gerar gráficos importantes durante o treinamento para analisar o comportamento dos modelos. Ao final do projeto, os alunos terão adquirido experiência prática na implementação de redes neurais.

Instruções

Identificador
Use o **número aleatório de 6 dígitos que você usou para o primeiro trabalho do curso** e escreva-o na primeira célula do notebook. Certifique-se de manter uma cópia desse número, pois ele será usado para fornecer o feedback.

Submissão
Envie seus arquivos através do ECLASS. Os arquivos enviados não podem ser substituídos por ninguém e não podem ser lidos por nenhum outro aluno. Você pode, no entanto, substituir seu envio quantas vezes quiser, reenviando, embora apenas a última versão enviada seja mantida.

Se você tiver problemas na última hora, envie um e-mail como anexo para alberto.paccanaro@fgv.br com o assunto "URGENTE – ENVIO DE TAREFA 3 ". NO corpo da mensagem, explique o motivo de não enviar através do ECLASS.

IMPORTANTE

- Seu envio consistirá em um único bloco de anotações Python implementando suas soluções.
- **O nome do arquivo será o número aleatório que o identifica (por exemplo, 568423.ipynb)**
- Este curso é composto por 2 partes. Certifique-se de que o código de ambas as partes está colocado nas células de código relevantes no notebook.
- NÃO ENVIE NENHUM CONJUNTO DE DADOS, apenas o código.
- Qualquer função auxiliar que você usará deve ser incluída no notebook – não envie scripts adicionais.

Todo o trabalho que você enviar deve ser exclusivamente seu próprio trabalho. As submissões de trabalhos do curso serão verificadas para isso.

Critérios de marcação

Este trabalho de curso é avaliado e obrigatório e vale 10% da sua nota final total para este curso. Para obter nota máxima para cada pergunta, você deve respondê-la corretamente, mas também completamente. Serão dadas notas para escrever código de estrutura de poço.

IMPORTANTE: Em geral, você pode usar *numpy* ou outras bibliotecas básicas para operações de matriz, mas você não pode usar qualquer função de biblioteca que implementaria alguns dos algoritmos que você é obrigado a implementar. Se você está em dúvida sobre uma função específica, envie-nos um e-mail.
Além disso, para o ajuste de parâmetros, você não tem permissão para usar qualquer função de pesquisa de "grid", como GridSearchCV do sklearn ou funções equivalentes.

TAREFA

Parte 1 – Regressão com Redes Neurais (valor: 50%)

Faça o download no ECLASS

- Tarefa_3_template.ipynb
- energy_efficiency.csv

Conjunto de dados e descrição do problema
Neste exercício, você usará o conjunto de dados de Predição de Eficiência Energética. Este conjunto de dados contém informações sobre a eficiência energética dos edifícios com base em oito características, incluindo o tamanho do edifício, a orientação e o tipo de materiais de construção utilizados. O conjunto de dados inclui dois alvos: carga de aquecimento e carga de resfriamento, que representam a energia necessária para aquecer e resfriar o edifício, respectivamente.
Este conjunto de dados é útil para a construção de redes neurais que predizem a eficiência energética de edifícios, o que é um problema importante no campo da energia sustentável. O conjunto de dados tem sido usado em vários artigos de pesquisa de aprendizado de máquina e fornece um problema de regressão desafiador.

Descrição do Exercício: Predição de Eficiência Energética com Redes Neurais
Neste exercício, você usará o conjunto de dados de Previsão de Eficiência Energética fornecido. Você construirá e treinará uma rede neural para prever a carga de aquecimento (coluna rotulada y1 no conjunto de dados) e a carga de resfriamento (coluna rotulada y2) de um edifício com base em suas características de eficiência energética.

Para concluir este exercício, você escreverá código para criar e treinar redes neurais para esse problema:

1. Divida o conjunto de dados em conjuntos de treinamento, validação e teste, usando uma proporção de 70:15:15.

2. Use numpy, construa uma rede neural que tome os recursos de eficiência energética como entrada e preveja o aquecimento e a carga de resfriamento como saída. Você escolherá o número de neurônios por camada e o número de camadas, mas cada camada deverá ter o mesmo número de neurônios.

3. Codifique o algoritmo de passagem direta e retropropagação para aprender os pesos da rede neural. Use o conjunto de treinamento para treinar a rede neural e atualizar os pesos usando a descida do gradiente estocástico. Você precisará regularizar sua rede neural usando decaimento dos pesos, ou seja, você incluirá um termo de regularização em sua função de erro.

4. Monitore o treinamento traçando as perdas de treinamento e validação ao longo das epochs.

O desempenho da sua rede neural será diferente dependendo do número de camadas, número de neurônios por camada e o valor de λ que controla a quantidade de decaimento de peso. Você experimentará com 3 valores de λ: 0 (sem decaimento de peso), 0.001 e 0.0001.
Para escolher a melhor configuração de rede e avaliar seu desempenho, você irá:

1. Calcular o loss (erro) para cada configuração no conjunto de validação.

2. Escolha uma das seguintes opções:

    a. Crie 3 redes para cada valor do parâmetro de regularização λ. A primeira rede com uma camada oculta e 100 neurônios, a segunda com duas camadas ocultas e 250 neurônios por camada, e a terceira com três camadas ocultas e 150 neurônios por camada. No final você deve ter 9 valores da loss no conjunto de validação.
    b. O número de camadas deve ser um argumento de entrada para a função que você está codificando (você deve usar uma estrutura de dados apropriada para armazenar as camadas ocultas). Gerar 3 heatmaps, um para cada valor do parâmetro λ de regularização, exibindo o erro no conjunto de validação e plotando o número de camadas e o número de neurônios em uma grade. Isso ajudará você a visualizar a melhor configuração para a rede neural. Para cada heatmap você pode escolher todas as nove combinações possíveis entre 1 a 3 camadas ocultas e 100, 150 e 250 neurônios por camada, mas você também pode escolher valores diferentes. **Esta opção dá-lhe 7.5 pontos de bónus.**
    *Note que para o ponto **a.** você pode ter uma variável W_x para cada camada ocultax, mas para o ponto **b.**, a fim de obter os pontos de bônus, você terá que ter um número variável de camadas e os pesos W_x devem ser armazenados em uma estrutura de dados apropriada de comprimento variável de acordo com o argumento de entrada.*

3. Treinar seu modelo final selecionando a melhor combinação de hiperparâmetros e avaliar o desempenho final da rede neural usando o conjunto de teste e a raiz do erro quadrático médio como métrica e relate isso.

Importante:

- Treine por 50 epochs.
- Defina a taxa de aprendizagem η como 0,01.

Parte 2 – Classificação com Redes Neurais (valor: 50%)

Faça o download dos dados do ECLASS

- CHD_49.csv

Descrição do conjunto de dados:
Este é um conjunto de dados do domínio médico. Descreve o problema do diagnóstico da doença cardíaca coronariana (DAC) através das abordagens da Medicina Tradicional Chinesa. Cada dado corresponde a um paciente representado por um conjunto de 49 características correspondentes à presença ou ausência de diferentes sintomas: sensação de frio ou calor, sudorese, etc. Os 6 rótulos representam a presença ou ausência de condições cardíacas específicas: deficiência da síndrome do qi cardíaco, deficiência da síndrome yang do coração, deficiência da síndrome do yin cardíaco, síndrome da estagnação do qi, síndrome do catarro turvo e síndrome da estase sanguínea.

Descrição do exercício: CHD49 Classificação multi-label com Redes Neurais
Neste exercício, você construirá e treinará uma rede neural para prever os 6 rótulos diferentes de CHD (últimas 6 colunas do conjunto de dados).

Para concluir este exercício, siga estes passos:

1. Carregue o conjunto de dados e divida-o em conjuntos de treinamento, validação e teste, usando uma proporção de 70:15:15.

2. Construa uma rede neural usando numpy que recebe os recursos como entrada e prevê os 6 rótulos diferentes. Você escolherá o número de neurônios por camada e o número de camadas, mas cada camada deverá ter o mesmo número de neurônios.

3. Codifique o algoritmo de passagem direta e retropropagação para aprender os pesos da rede neural. Use o conjunto de treinamento para treinar a rede neural e atualizar os pesos usando a descida do gradiente batch. Você precisará regularizar sua rede neural usando decaimento dos pesos, ou seja, você incluirá um termo de regularização em sua função de erro.

4. Monitore o treinamento traçando as perdas de treinamento e validação ao longo das epochs.

O desempenho da sua rede neural será diferente dependendo do número de camadas, número de neurônios por camada e o valor de λ que controla a quantidade de decaimento de peso. Você experimentará com 3 valores de λ: 0 (sem decaimento de peso), 0.1 e 0.01.
Para escolher a melhor configuração de rede e avaliar seu desempenho, você irá:

1. Calcular o loss (erro) para cada configuração no conjunto de validação.

2. Escolha uma das seguintes opções:

    a. Crie 3 redes para cada valor do parâmetro de regularização λ. A primeira rede com uma camada oculta e 100 neurônios, a segunda com duas camadas ocultas e 250 neurônios por camada, e a terceira com três camadas ocultas e 150 neurônios por camada. No final você deve ter 9 valores da loss no conjunto de validação.
    b. O número de camadas deve ser um argumento de entrada para a função que você está codificando (você deve usar uma estrutura de dados apropriada para armazenar as camadas ocultas). Gerar 3 heatmaps, um para cada valor do parâmetro λ de regularização, exibindo o erro no conjunto de validação e plotando o número de camadas e o número de neurônios em uma grade. Isso ajudará você a visualizar a melhor configuração para a rede neural. Para cada heatmap você pode escolher todas as nove combinações possíveis entre 1 a 3 camadas ocultas e 100, 150 e 250 neurônios por camada, mas você também pode escolher valores diferentes. **Esta opção dá-lhe 7.5 pontos de bónus.** *Note que para o ponto **a.** você pode ter uma variável W_x para cada camada ocultax, mas para o ponto **b.**, a fim de obter os pontos de bônus, você terá que ter um número variável de camadas e os pesos W_x devem ser armazenados em uma estrutura de dados apropriada de comprimento variável de acordo com o argumento de entrada.*

1. Treine seu modelo final selecionando a melhor combinação de hiperparâmetros e avalie o desempenho final da rede neural usando o conjunto de teste e calculando a área sob a curva ROC, a precisão e o score F1 como métricas e relate isso.

Importante:

- Treine por 1000 epochs.
- Defina a taxa de aprendizagem η como 0,01.