Universidade Federal de Minas Gerais
Disciplina de Computação Natural
Prof. Gisele L. Pappa

Trabalho Prático 2
Redes neurais aplicadas a problemas de classificação

Data de Entrega: 28/11/2016
Atenção: devido ao calendário do fim do semestre, não serão aceitos trabalhos entregues após 30/11.

---

1 Introdução

Tarefas de classificação estão entre os problemas de decisão que aparecem com mais frequência no nosso cotidiano. Tais tarefas surgem quando temos objetos que devem ser associados a grupos pré-determinados, tomando como base um certo número de atributos relacionados a esses objetos. Dentro do contexto de aprendizado de máquina, classificação é uma técnica de mineração de dados utilizada para prever, a partir de instâncias cuja classe é conhecida, a classe de novas instâncias.

Uma forma muito utilizada para resolver problemas de classificação envolve o uso de redes neurais. Nessa direção, o objetivo deste trabalho é colocar em prática conceitos relacionados às redes neurais. Para isso, deve-se implementar uma rede capaz de resolver dois problemas de classificação. O primeiro desses problemas está relacionado à determinação do resultado de Jogos da Velha a partir do conjunto de possíveis combinações ao final das partidas, enquanto o segundo envolve a classificação de veículos a partir de atributos como preço e tamanho. O uso de bibliotecas que facilitem a implementação das redes está liberado. Essa liberação, muito mais do que facilitar o trabalho a ser feito, visa concentrar o foco de atenção para a análise experimental. Espera-se, portanto, uma análise completa e bem feita. O nível de exigência durante a avaliação será, consequentemente, bem maior do que o que foi tido no primeiro trabalho prático.

2 Bibliotecas

Diversas bibliotecas oferecem suporte para implementação de redes neurais. Esta seção lista algumas à título de recomendação, escolhidas por sua facilidade de utilização. Este trabalho, no entanto, pode ser realizado utilizando outra biblioteca, desde que sejam seguidos os passos de experimentação explicados na Seção 4. Ferramentas como o Weka ou Orange não devem ser utilizadas. Se você for escolher qualquer outra biblioteca por motivos de linguagem, por favor, nos avise antes.

As bibliotecas sugeridas são:

– TensorFlow - Biblioteca em Python com código aberto mantida pela Google.
– NeuroLab - Simples e poderosa biblioteca de redes neurais para Python, de acordo com os criadores.
– PyBrain - Contém algoritmos para redes neurais, aprendizagem por reforço e aprendizagem não supervisionada.
– NeuralNet - Biblioteca em R construída com foco em redes neurais.

3 Bases de dados

Duas bases de dados serão utilizadas neste trabalho, ambas obtidas no *Irvine Machine Learning Repository* [1], que é mantido pela Universidade da Califórnia. A primeira dessas bases está relacionada a uma tarefa de classificação binária, onde os atributos de entrada codificam o conjunto completo de configurações possíveis para estados finais do Jogo da Velha, onde assume-se que o "x" jogou primeiro. A classe de cada instância é denominada "vitória do x" (ou seja, verdadeiro quando "x" consegue uma das oito diferentes formas de se criar uma linha com três símbolos "x"). A Tabela 1 mostra os possíveis valores para cada um dos atributos da base. O caractere "b" indica que a posição chegou ao final do jogo sem ser preenchida. A Tabela 2 mostra a distribuição de classes para a mesma base.

	Atributo	Possíveis valores
1	*top-left-square*	x, o, b
2	*top-middle-square*	x, o, b
3	*top-right-square*	x, o, b
4	*middle-left-square*	x, o, b
5	*middle-middle-square*	x, o, b
6	*middle-right-square*	x, o, b
7	*bottom-left-square*	x, o, b
8	*bottom-middle-square*	x, o, b
9	*bottom-right-square*	x, o, b
10	*Class*	*positive*, *negative*

Tabela 1: Valores dos atributos para a base de dados "*Tic-Tac-Toe Endgame*"

Classe	Número de instâncias
positive	626 (65,3%)
negative	332 (34,7%)
Total	958

Tabela 2: Distribuição de classes para a base de dados "*Tic-Tac-Toe Endgame*"

A segunda base a ser utilizada neste trabalho também envolve uma tarefa de classificação, porém com quatro classes. Nela, carros são avaliados com base em seus preços e características técnicas. A classe de cada instância determina o nível de aceitabilidade de cada veículo. As Tabelas 3 e 4 mostram os possíveis valores para cada um dos atributos e a distribuição de classes da base.

	Atributo	Possíveis valores	Descrição
1	*buying*	*vhigh*, *high*, *med*, *low*	Preço de compra
2	*maint*	*vhigh*, *high*, *med*, *low*	Custo de manutenção
3	*doors*	2, 3, 4, *5more*	Número de portas
4	*persons*	2, 4, *more*	Capacidade de passageiros
5	*lug_boot*	*small*, *med*, *big*	Tamanho do porta-malas
6	*safety*	*low*, *med*, *high*	Segurança estimada do carro
7	*Class*	*unacc*, *acc*, *good*, *vgood*	Aceitabilidade do veículo

Tabela 3: Valores dos atributos para a base de dados "*Car*"

Classe	Número de instâncias
unacc	1210 (70,02%)
acc	384 (22,22%)
good	69 (3,99%)
v-good	65 (3,76%)
Total	1728

Tabela 4: Distribuição de classes para a base de dados "*Car*"

4 Metodologia Experimental

Como mencionado, o principal objetivo desse trabalho é entender a sensibilidade da rede em relação aos valores de seus parâmetros. As bases de dados consideradas são pequenas exatamente para que você possa brincar com os parâmetros em um intervalo de tempo compatível com o do trabalho.

Guia para a implementação inicial

- Escolha uma biblioteca que auxilie na implementação da rede neural. É necessário informar a biblioteca escolhida, entretanto não é indispensável explicar o porquê da escolha. Antes de começar o processo experimental, faça uma boa pesquisa sobre as vantagens, peculiaridades e limitações de pelo menos parte delas, de forma a evitar retrabalhos.
- Escolha um tipo de rede neural (Perceptron, RBF, etc.)
- Analise se o formato das bases de dados fornecidas é compatível com o formato de entrada da biblioteca escolhida. Analise também se serão necessárias transformações nos dados de entrada. Atributos discretos, por exemplo, devem ser representados de forma apropriada. Informações sobre modificações nos dados de entrada podem ser encontradas nos slides da aula 18, onde foram discutidos diversos tipos de transformações.
- Estabeleça uma estrutura inicial para a rede, o que significa definir como será a entrada e a saída e quantos neurônios serão utilizados para representá-las. Se essa estrutura inicial já contar com uma camada escondida, o número de neurônios dessa

camada também deve ser definido. Para maiores informações sobre como representar a saída da rede para problemas de classificação, veja os slides da aula 18.

- Estabeleça parâmetros iniciais para a rede. A lista de decisões que devem ser tomadas incluem:
  - Inicialização dos pesos;
  - Escolha da taxa de aprendizagem;
  - Função de ativação a ser utilizada.

  Consulte os slides da disciplina e/ou as FAQ de redes neurais para ver os valores padrão para iniciar o processo de escolha de parâmetros.

- Os dados utilizados devem ser divididos em duas ou três partes, dependendo do critério de parada selecionado. Quando os dados são divididos em duas partes, a primeira (que normalmente corresponde a 70% do total) é utilizada para treinar a rede por um número máximo de épocas. Ao fim do treinamento, a segunda parte é utilizada para medir a capacidade de generalização da rede.

  Quando os dados são divididos em três partes, a primeira parte é novamente utilizada para treinar a rede, enquanto a segunda é utilizada para validá-la, isto é, medir seu erro. Nesse caso, o treinamento pode parar quando a rede atingir um erro mínimo nesse segundo conjunto de dados, chamado conjunto de validação. A terceira parte dos dados é utilizada apenas ao fim do treinamento, para medir a capacidade de generalização da rede.

  Independente da abordagem utilizada, o ideal é que um procedimento de validação cruzada seja utilizado para garantir que os resultados de generalização não estejam sendo obtidos ao acaso. Uma validação cruzada de 10 partições é um ideal para os problema que estamos tentando resolver. Mais informação sobre sobre esse processo de separação de dados podem ser encontrados nos slides da disciplina.

Guia para o treinamento

- O primeiro teste a ser feito envolve a forma de se treinar a rede. Como foi explicado em sala, existem duas formas: (i) online, onde os pesos são atualizados logo após a apresentação de um exemplo à rede e (ii) batch, onde os pesos são atualizados ao final de cada época. Em teoria a primeira opção leva a convergências mais rápidas, enquanto a segunda à taxas de erros menores. Varie essa forma de treinamento e verifique se aconteceu o que era esperado. Comente os resultados.

- O segundo teste envolve a definição de um critério de parada para a rede. Normalmente tal critério depende da convergência da rede, que acontece quando a diferença absoluta entre os erros de uma época para outra é suficiente pequena. Determine um critério de parada coerente para a sua implementação.

- Outro aspecto que deve ser analisado se refere à generalização da rede. No nosso caso o número de exemplos escolhidas para a etapa de treinamento e a estrutura escolhida tendem a impactar a capacidade de generalização da rede. Realize testes com diferentes números de instâncias de treinamento e verifique qual a configuração ideal para a rede. Tente também analisar quais mudanças na estrutura da rede reduzem a ocorrência de *overfitting*.

– Todos os resultados reportados devem basear-se no erro obtido pela rede tanto na fase de treinamento quanto na fase de teste, para que seja possível realizar uma análise de *overfitting*.

– Realize testes envolvendo variações nos parâmetros que foram definidos durante a implementação inicial da rede (inicialização dos pesos, taxa de aprendizagem, número de neurônios e camadas escondidas e funções de ativação utilizadas). Tente mostrar de forma clara qual o impacto da variação de cada um desses parâmetros.

O que deve ser entregue...

– Quaisquer arquivos utilizados durante a implementação da rede.

– Documentação do trabalho:

  – Introdução.
  – Descrição do processo de implementação da rede, incluindo detalhes da representação utilizada por você e/ou pela biblioteca escolhida.
  – Análise experimental detalhada, explicando a forma como foram feitas as escolhas iniciais para os parâmetros da rede e conjunto de testes realizados para otimizar os resultados obtidos (sempre variando apenas um parâmetro por vez).
  – Conclusões.
  – Bibliografia.

A entrega DEVE ser feita pelo Moodle na forma de um único arquivo zipado, contendo os arquivos relacionados à implementação da rede e a documentação do trabalho.

Considerações Finais

– Lembrem-se que ao mexer em um dos parâmetros, todos os outros devem ser mantidos constantes, e que a análise dos parâmetros é de certa forma interativa. A configuração de parâmetros raramente vai ser ótima, mas pequenos testes podem melhorar a qualidade das soluções encontradas.

– Por ser um método estocástico, a avaliação experimental do algoritmo baseado em redes neurais deve ser realizada com repetições, de forma que os resultados possam ser reportados segundo o valor médio obtido e o respectivo desvio-padrão. A realização de 30 repetições pode ser um bom ponto de partida (lembrando que desvio-padrão alto sugere um maior número de repetições).

Referências

[1] Lichman, M. (2013). *UCI Machine Learning Repository* [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.