

Classificação de Sinais de Sonar Passivo utilizando Redes Neurais Artificiais

Laboratório de Processamento de Sinais - DEL

Lucas de Andrade Cerqueira José Manoel de Seixas

11 de Maio de 2016

Resumo

Projeto em parceria com a Marinha do Brasil com o objetivo de classificar sinais provenientes de navios através de técnicas de aprendizado de máquina. O modelo de rede escolhido foi rede neural artificial com muitas camadas, também conhecido como Deep Learning. Após o treinamento adequado utilizando gravações de navios cedidas pela Marinha, é possível identificar a qual classe pertencem e detectar possíveis novidades. O trabalho consiste em encontrar a melhor forma de treinar a arquitetura de rede neural com muitas camadas a fim de obter o menor erro possível nas classificações.

Introdução

Para um submarino, o mais importante é não denunciar sua posição. Ao usar um sonar ativo, o qual emite ondas sonoras que, ao colidirem com obstáculos, retornam ao emissor, o submarino poderia ser localizado por navios inimigos. Já em um sonar passivo, o submarino está sempre captando sinais sonoros e cabe a um operador extrair qual a classe de navio que produziu o ruído captado pelo sonar passivo.

Contudo, a análise desse som é realizada no domínio da frequência e o ambiente marinho é altamente ruidoso. O ruído próprio e o ruído de fundo dificultam a identificação dos contatos. Por esse motivo, um sistema de apoio para a classificação desses contatos pode auxiliar o operador sonar em seu trabalho.

Objetivo

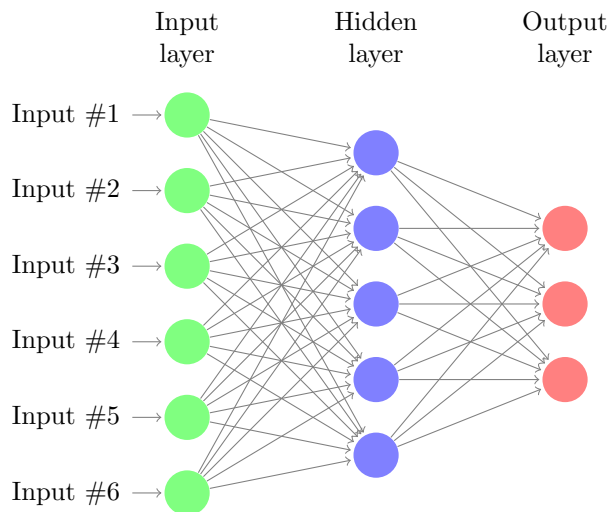
O projeto tem por objetivo a classificação de sinais de sonar passivo através do uso de redes neurais artificiais, especificamente utilizando o modelo de *Deep Learning* com *Stacked Autoencoders* e analisando o desempenho do mesmo.

Metodologia/Teoria

Redes neurais artificiais

Redes neurais artificiais são uma família de métodos inspirada em modelos biológicos do cérebro capazes de aprender características e reconhecer padrões, sendo amplamente utilizadas para a classificação de sinais em sonar passivo. Geralmente, são apresentadas como neurônios conectados entre si com um valor numérico para essas conexões, conhecido como peso.

O processo de aprendizado ocorre através do treinamento, que pode ser supervisionado ou não supervisionado: no primeiro, os eventos a serem usados no treinamento precisam ser etiquetados com relação a que classe pertencem; no treinamento não supervisionado, não ocorre essa etiquetagem.



Exemplo de rede neural artificial com 6 neurônios na entrada, 5 na camada escondida e uma camada de saída com 3 neurônios.

Stacked Autoencoder (SAE)

Um *Stacked Autoencoder* é uma rede neural com o objetivo de aprender representações de um conjunto de dados e é utilizado, geralmente, para redução de dimensionalidade e, mais recentemente, para aprender modelos geradores de dados.

Sua arquitetura é baseada na conexão entre *autoencoders*, onde cada um é formado por um *encoder* e por um *decoder*. Ao passar pelo *encoder*, a dimensão da entrada é reduzida e, ao passar esse resultado pelo *decoder*, retorna para uma dimensão de mesmo tamanho da entrada do *autoencoder*.

O treinamento dessa arquitetura é realizado com um algoritmo conhecido como *layer-wise pretraining* [1]. É não-supervisionado e, como o próprio nome sugere, é um pre-treinamento que ocorre em cada camada de *autoencoder*. Inicia-se treinando o primeiro *autoencoder* e, após treiná-lo, seus pesos são fixados e sua saída será utilizada como entrada para o treinamento do próximo. Após esse processo ser realizado até a última camada, pode ser feito um *fine-tuning* de todos os pesos do SAE utilizando alvos conhecidos e realizando uma retropropagação do erro **alvo-saída**, conhecido como *backpropagation*.

Essa retropropagação do erro também é utilizada em redes neurais comuns. Porém, ao criar redes mais profundas, há uma tendência à convergência dos pesos das camadas mais próximas à saída mais rápida do que de pesos mais próximos à entrada. Com isso, redes mais profundas apresentam uma falha de generalização se treinadas somente com *backpropagation*, tornando o treina-

mento ineficaz para classificação. O *layer-wise pretraining* tem o objetivo de iniciar os pesos da estrutura da rede neural a partir de valores que favoreçam a minimização do erro. Um outro ponto relevante desse pré-treinamento é uma possível extração de características do conjunto de dados [2].

Dados utilizados

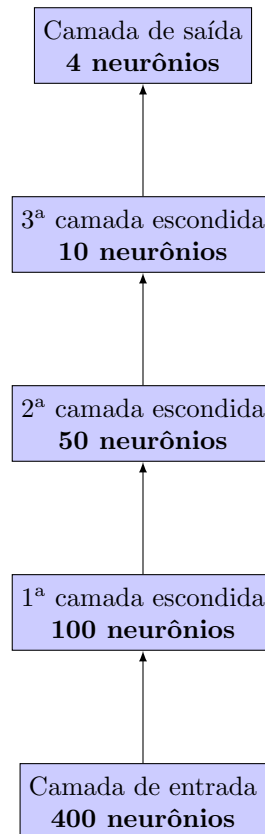
O projeto segue a mesma aplicação para classificação de padrões adotada pela Marinha do Brasil. Os dados de entrada utilizados são gravações, cedidas pela Marinha do Brasil, que foram realizadas na raia acústica de Arraial do Cabo. Ao todo, foram gravadas as corridas de quatro classes de navios, detalhadas abaixo:

- **Classe 1:** 5 corridas, totalizando 4313 janelas
- **Classe 2:** 10 corridas, totalizando 9785 janelas
- **Classe 3:** 9 corridas, totalizando 3837 janelas
- **Classe 4:** 10 corridas, totalizando 7921 janelas

Essas gravações são arquivos *.wav* que são pré-processados e passam por uma análise LOFAR. Dessa análise, são extraídos 400 *bins* de frequência que serão a entrada do SAE.

Estrutura utilizada no SAE

Para este projeto, está sendo usada a estrutura descrita a seguir:



Até o momento, não foi buscada a estrutura ótima do SAE para o conjunto de dados e a finalidade pretendida. Ainda é preciso fazer testes exaustivos para determinar a quantidade de neurônios em cada camada e quantas camadas devem ser utilizadas.

Resultados

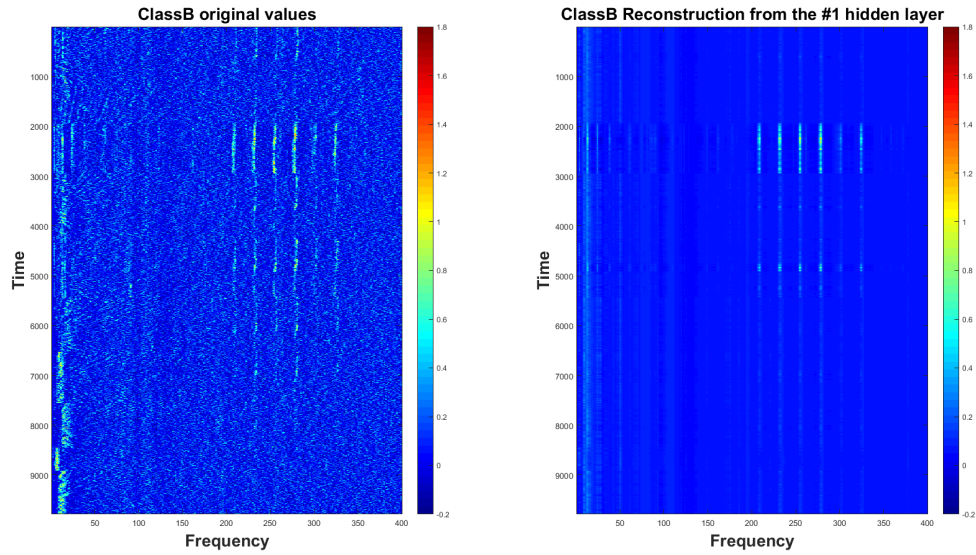


Figura 1: Classe 2 - Entrada original e reconstrução

O gráfico acima mostra um exemplo da tendência do primeiro autoencoder extrair características dos dados de entrada. Para a **classe 2**, a reconstrução da entrada realçou bins de frequência que estavam bem nítidos na entrada.

A seguir, seguem duas matrizes de confusão, que traduzem o desempenho da rede em classificar as entradas com relação aos alvos definidos para cada uma delas:

Confusion matrix for training set					
Output Class	1	2	3	4	
	<div>3680 15.8%</div>	<div>66 0.3%</div>	<div>24 0.1%</div>	<div>11 0.0%</div>	<div>97.3% 2.7%</div>
	<div>132 0.6%</div>	<div>8559 36.8%</div>	<div>161 0.7%</div>	<div>83 0.4%</div>	<div>95.8% 4.2%</div>
	<div>57 0.2%</div>	<div>83 0.4%</div>	<div>3259 14.0%</div>	<div>23 0.1%</div>	<div>95.2% 4.8%</div>
	<div>13 0.1%</div>	<div>99 0.4%</div>	<div>9 0.0%</div>	<div>7012 30.1%</div>	<div>98.3% 1.7%</div>
	1	2	3	4	
	<div>94.8% 5.2%</div>	<div>97.2% 2.8%</div>	<div>94.4% 5.6%</div>	<div>98.4% 1.6%</div>	<div>96.7% 3.3%</div>
	Target Class				

Figura 2: Matriz de confusão para o conjunto de treinamento

Fazendo uma análise da matriz de confusão acima extraída de eventos do conjunto de treinamento, pode-se observar que os desempenhos por classe ficaram bem próximos entre si:

- 94,8% dos eventos etiquetados como Classe 1 foram corretamente classificados
- 97,2% dos eventos etiquetados como Classe 2 foram corretamente classificados
- 94,4% dos eventos etiquetados como Classe 3 foram corretamente classificados
- 98,4% dos eventos etiquetados como Classe 4 foram corretamente classificados

Outro resultado a se avaliar é a confiabilidade da classificação:

- 97,3% dos eventos classificados como Classe 1 eram de fato da Classe 1
- 95,8% dos eventos classificados como Classe 2 eram de fato da Classe 2
- 95,2% dos eventos classificados como Classe 3 eram de fato da Classe 3
- 98,3% dos eventos classificados como Classe 4 eram de fato da Classe 4

Além disso, dentre o total de eventos, 96,7% foram classificados corretamente.

Confusion matrix for test set					
Output Class	1	2	3	4	
	365 14.1%	29 1.1%	15 0.6%	32 1.2%	82.8% 17.2%
	35 1.4%	859 33.2%	74 2.9%	13 0.5%	87.6% 12.4%
	18 0.7%	71 2.7%	274 10.6%	26 1.0%	70.4% 29.6%
	13 0.5%	19 0.7%	21 0.8%	721 27.9%	93.2% 6.8%
	84.7% 15.3%	87.8% 12.2%	71.4% 28.6%	91.0% 9.0%	85.8% 14.2%
Target Class					
1	2	3	4		

Figura 3: Matriz de confusão para o conjunto de teste

Analisando da mesma forma os resultados do conjunto de teste, observa-se que o desempenho na classificação da classe 3 foi bem inferior ao das outras classes:

- 84,7% dos eventos etiquetados como Classe 1 foram corretamente classificados
- 87,8% dos eventos etiquetados como Classe 2 foram corretamente classificados
- 71,4% dos eventos etiquetados como Classe 3 foram corretamente classificados
- 91,0% dos eventos etiquetados como Classe 4 foram corretamente classificados

Analisando a confiabilidade da classificação:

- 82,8% dos eventos classificados como Classe 1 eram de fato da Classe 1
- 87,6% dos eventos classificados como Classe 2 eram de fato da Classe 2
- 70,4% dos eventos classificados como Classe 3 eram de fato da Classe 3
- 93,2% dos eventos classificados como Classe 4 eram de fato da Classe 4

Dentre o total de eventos, 85,8% foram classificados corretamente. A Classe 3 pode estar sendo confundida com outras classes, especialmente a Classe 2, já que 2,9% dos eventos da Classe 3 foram classificados erroneamente como Classe 2 e, dos eventos classificados como Classe 3, 2,7% pertenciam à Classe 2.

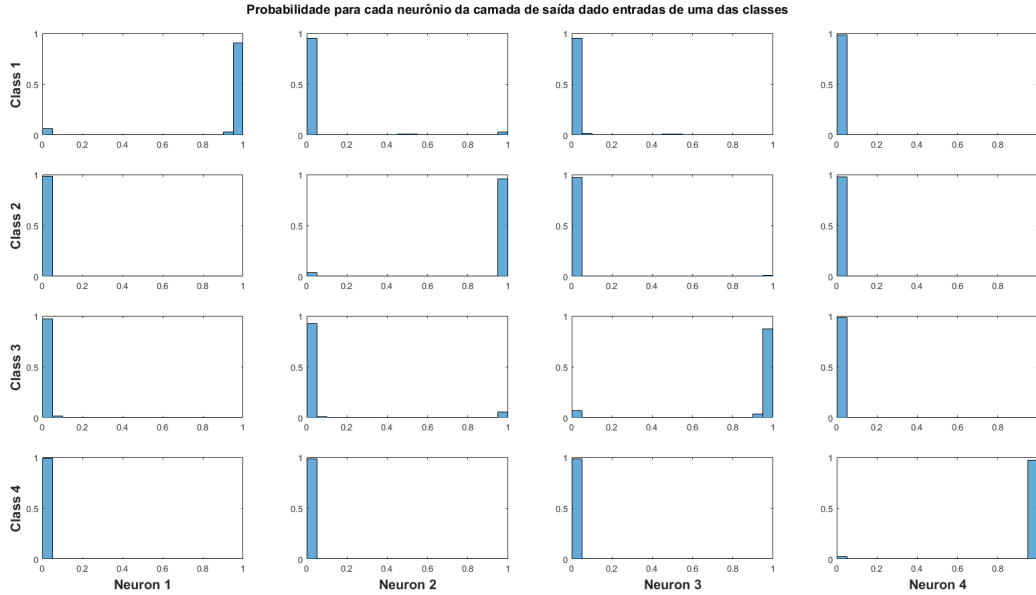


Figura 4: Distribuições na camada de saída por classe e por neurônio

Pelo gráfico acima, observamos o comportamento esperado dos neurônios da camada de saída. A distribuição de saída encontra-se saturada nos pontos de treinamento, mostrando que a extração de características (feita pelas camadas mais próximas à entrada) tende a aumentar o poder discriminatório das entradas.

A etiquetagem dos eventos por classe foi feita de forma que somente um neurônio da saída ficasse ativo para uma classe específica. O neurônio 1 é ativado para eventos da classe 1, o neurônio 2 é ativado para eventos da classe 2 e sucessivamente. Além disso, essas distribuições corroboram com o resultado das matrizes de confusão, pois eventos da classe 3 excitaram o neurônio 2, podendo levar a uma classificação errônea.

Conclusões

Analisando os resultados acima, o *Stacked Autoencoder* obteve um bom desempenho para a classificação dos sinais de navios cedidos pela Marinha e a extração de características da entrada pôde ser confirmada através da reconstrução do primeiro *autoencoder* do SAE. A classe C obteve um desempenho inferior ao das outras, porém o fato das classes não possuírem o mesmo número de eventos pode favorecer o treinamento de uma classe em detrimento de outra.

Por fim, mais testes serão realizados para se chegar à melhor estrutura para o SAE e para conseguir uma extração de características da entrada que leve a um melhor treinamento de classificação.

Bibliografia

- [1] Yoshua Bengio, Pascal Lamblin, Dan Popovici e Hugo Larochelle (2006). Greedy Layer-Wise Training of Deep Networks. *Dept.IRO, Université de Montréal*.
- [2] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio e Pierre-Antoine Manzagol (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning*
- [3] Yoshua Bengio, Yann LeCun (2009). Tutorial: Learning Deep Architectures. *ICML Workshop on Learning Feature Hierarchies*.
- [4] G.E. Hinton, R.R. Salakhutdinov (2006). Reducing the Dimensionality of Data with Neural Networks *Science, 28 Jul 2006*
- [5] http://ufldl.stanford.edu/wiki/index.php/Stacked_Autoencoders

Comentários finais

Avaliação feita pelo bolsista

Sem dúvida, essa iniciação científica só me acrescentou coisas boas. Estou estudando um assunto pelo qual me interessa muito e que está em alta, conheci pessoas brilhantes no ambiente de trabalho e amadureci muito no âmbito de pesquisa. Pretendo continuar no projeto buscando resultados ainda melhores e só tenho a agradecer ao professor Seixas e a um de seus alunos de doutorado, Natanael Júnior, que estão me auxiliando muito nesse projeto.