

# PREDIÇÃO DE JOGADAS DE FUTEBOL AMERICANO UTILIZANDO REDES NEURAIIS

DIEGO DA SILVA RODRIGUES\*

\*Laboratório de Processamento de Sinais

COPPE/Poli - UFRJ

Rio de Janeiro, RJ, Brasil

Email: [diegosrx@gmail.com](mailto:diegosrx@gmail.com)

**Resumo**— Este artigo apresenta os resultados da aplicação de redes neurais para a predição de jogadas de futebol americano. Na partida, o ataque deve avançar em campo com quatro tentativas, optando por uma jogada de passe ou corrida. Para cada uma das tentativas, uma rede neural foi treinada utilizando atributos da jogada, como posição no campo, tempo de jogo, placar, etc. O objetivo é prever se acontecerá um passe ou uma corrida. A taxa de acerto para a primeira tentativa foi de 61,39%, para a segunda foi de 72,99% e para a terceira foi de 94,41%. O classificador de jogadas pode ser utilizado por softwares de inteligência artificial para jogos, simuladores de comportamento de determinada equipe ou para inferência sobre qual será jogada mais provável durante uma partida.

**Palavras-chave**— Redes Neurais, Classificação, Previsão de jogada de Futebol Americano.

## 1 Introdução

O futebol americano é um esporte praticado principalmente nos Estados Unidos, sendo pouco conhecido no Brasil e praticado apenas em campeonatos amadores regionais. Além de ser um esporte conhecido pelo contato físico intenso, sua principal característica é a estratégia.

O campo de futebol americano tem 100 jardas (1 jarda = 0.9144 metros) de comprimento, mais duas endzones (zonas de fundo) de 10 jardas, totalizando 120 jardas. No final de cada uma das endzones, há uma trave de metal em forma de “Y”, chamado de *field goal*. O objetivo dos times é pontuar atravessando ou recebendo um passe dentro da endzone, realizando um *touchdown*, ou chutando a bola através do “Y”, marcando um *field goal*. O *touchdown* vale seis pontos e o *field goal* vale três. Cada equipe conta com três sub-equipes: o ataque, responsável por tentar avançar dentro do campo e pontuar; a defesa, que deve impedir o avanço do ataque adversário; o time especial, que entra em campo em situações de troca de posse de bola e chute.

No início da partida, os times especiais se enfrentam no *kick-off* inicial, onde um time chuta a bola para o outro time retorná-la. O time de chute então tenta impedir o avanço do retornador - jogador que recebeu o chute adversário - e o time de retorno tenta abrir passagem para ele retornar o maior número possível de jardas. A jogada encerra-se quando o retornador é derrubado, dando início ao drive de ataque.

A bola é posicionada no centro do campo, paralelamente ao lugar onde o retornador foi derrubado ou saiu de campo. O time de ataque entra em campo, com o objetivo de avançar a bola rumo à endzone adversária. O ataque tem quatro tentativas (*downs*) para avançar dez jardas do campo, obtendo um *first-down* (mais quatro tentativas).

A jogada tem início quando o *Quarterback* recebe a bola do *Center* no snap (movimentação onde o *Center* entrega a bola por debaixo das pernas para o *Quarterback*) e realiza a jogada, entregando a bola para um *Running Back* (corredor) para que ele possa avançar correndo com a bola ou arremessando para um *Wide-Receiver* (recebedor). Analogamente, o time de defesa deve derrubar o jogador que recebe a bola do *Quarterback* ou o próprio, impedindo o avanço do time de ataque no campo.

O jogo é disputado em dois tempos, cada um dividido em dois quartos de quinze minutos. Cada time conta com três *Time-outs* (tempos técnicos) para pedir em cada tempo de jogo. A contagem de tempo (pausas no relógio) tem um formato especial nos dois minutos finais de cada tempo, permitindo aos times manipular a fluência do tempo de jogo (*two-minute warning*). Ao final do jogo, vence o time que tiver realizado o maior número de pontos. Mais informações sobre a regra do esporte podem ser encontradas em (NFL, 2008).

O objetivo é prever se o time de ataque optará por uma jogada de passe ou de corrida. Um passe permite ao time de ataque normalmente avançar mais jardas numa só jogada, com o risco do seu *Quarterback* ser derrubado ou a bola ser interceptada pelo time de defesa. Uma corrida em geral avança menos jardas que um passe, mas a bola permanece “protegida” nos braços de um jogador do time de ataque do começo ao fim da jogada.

Este compromisso entre passe e corrida gera momentos de jogo onde há possibilidade maior de determinada escolha: um time perdendo, no final de jogo, precisa fazer vários passes para avançar no campo o mais rápido possível; um time que precise avançar apenas uma jarda no campo pode fazê-lo de forma mais garantida correndo com a bola. Fora esses momentos característicos, a opção por passe ou corrida é afetada por caracterís-

ticas de jogo, do elenco do time de ataque e da preferência do *Quarterback* e coordenador de ataque. Logo, dadas as características que ilustram o momento de jogo para uma jogada, a meta é encontrar quais destas características são as mais relevantes e prever se será um passe ou uma corrida.

Como o objetivo é prever se determinada jogada será um passe ou uma corrida, o classificador usando redes neurais pode ser utilizado em simuladores do esporte, como por exemplo, inteligência artificial de jogos. Outra aplicação seria o auxílio a decisão durante o jogo, onde a rede neural realizaria uma inferência sobre a possibilidade de uma determinada jogada. O classificador foi concebido com esta função, e tal modelo requisitaria o treinamento da rede para cada *quarterback* diferente para ser utilizado na prática. Dado que a rede foi treinada com a base de dados de jogadas de um determinado *quarterback*, uma terceira aplicação seria simular o comportamento de tal equipe, para auxílio nos treinamentos pré-jogo, por exemplo.

Neste trabalho, uma rede neural foi treinada para cada *down*, totalizando três. Esta abordagem é uma extensão de (Rodrigues, 2009), onde apenas uma rede neural era utilizada e o número de *downs* era um dos atributos. Outras duas implementações utilizando árvores de decisão através do algoritmo C4.5 (Quinlan, 1994) e ID3 (Quinlan, 1986) serviram como estudo prévio do problema e o resultado obtido pelos dois métodos foi comparado ao resultado obtido pelas redes neurais. Em (Pines, 2005), os classificadores foram construídos utilizando regressão logística, numa abordagem parecida com a deste trabalho. Este foi o único trabalho encontrado ligado ao tema de predição de jogadas de futebol americano.

## 2 Metodologia

Nesta seção será abordada qual foi a metodologia utilizada para o treinamento. Como os atributos foram modelados, qual foi a arquitetura de rede utilizada e qual foi o banco de dados utilizado para treino e teste das redes.

### 2.1 Modelagem dos atributos

As características da jogada (atributos) podem ser divididas em dois grupos: fatores gerais da jogada e fatores circunstanciais do jogo.

No grupo de fatores gerais da jogada estão inclusos: qual é a tentativa; quantas jardas faltam para avançar dez jardas; qual a posição no campo; qual o tempo de jogo (quarto, minutos corridos, se está nos dois minutos finais); qual a formação que o time de ataque está utilizando (como está distribuída a equipe: quantos são os recebedores, quantos corredores, etc).

No grupo de fatores circunstanciais do jogo estão inclusos: qual o placar atual; Qual foi a jo-

gada anterior; condições climáticas (vento, chuva); quantos *time-outs* ainda podem ser pedidos; qual a produtividade do jogo corrido até agora (média de jardas alcançadas, jardas máximas alcançadas); produtividade do jogo de passes e muitos outros. Todos os atributos listados na tabela 1 foram uti-

Tabela 1: Atributos utilizados no treinamento das redes.

Nº	Atributo	Tipo
1	Distância para o <i>first-down</i>	Contínuo
2	Posição no campo	Contínuo
3	Diferença no placar	Contínuo
4	Quarto de jogo	Discreto
5	Tempo	Contínuo
6	<i>Two-Minute Warning</i>	Binário
7	Última jogada - Passe	Binário
8	Última jogada - Corrida	Binário
9	Última jogada - Especial	Binário
10	Última jogada - Jardas	Binário

lizados no treinamento das redes.

### 2.2 Base de dados

Os dados utilizados neste trabalho são todas as jogadas dos 10 primeiros jogos do time da NFL Indianápolis Colts na temporada de 2006. O *quarterback* deste time, Payton Manning, é um dos melhores da liga americana e é conhecido pela sua capacidade de modificar as jogadas dentro de campo, dependendo da configuração de defesa adversária. O banco de dados conta com 298 jogadas de passe e 233 jogadas de corrida. O resumo de quantos vetores existem para cada tentativa está na tabela 2. O banco de dados foi gerado a partir da informação do *Play-by-Play* de cada jogo, que é a descrição da sequência de jogadas, obtido no site da *National Football League* (N.F.L.) (NFL, 2006).

Tabela 2: Número de padrões dividido pelo número de tentativa e pelas duas classes, passe e corrida.

Tentativa	Passe	Corrida
Primeiro <i>down</i>	113	135
Segundo <i>down</i>	100	86
Terceiro <i>down</i>	85	12
Total	298	233

### 2.3 Arquitetura da rede

A rede escolhida foi a MLP (*Multilayer Perceptron*) com uma camada escondida (Haykin, 2007), utilizando o algoritmo de treinamento supervisionado *resilient backpropagation* (Riedmiller, 1994). O modelo foi implementado utilizando o *toolbox* de redes neurais do *MATLAB*. Todos os neurônios da rede utilizam a função de ativação tangente hiperbólica.

No total, três redes neurais foram treinadas. Cada uma delas foi utilizada para classificar as jogadas para um dos três *downs*. A primeira parte

do treinamento consistiu em escolher o número de neurônios na camada oculta. Para isto, foi utilizada busca incremental exaustiva. As redes foram treinadas variando o número de neurônios na camada oculta de dois até dez. Caso a taxa de acerto ainda estivesse aumentando com o número de neurônios, mais dez treinamentos eram realizados até a taxa de acerto começar a diminuir.

Para validar cada um dos treinamentos, foi utilizada validação cruzada 60-20-20. Para cada uma das redes (primeira, segunda e terceira descida), os vetores de passe e corrida foram divididos em cinco conjuntos, totalizando dez (cinco conjuntos com vetores de passe, cinco com vetores de corrida). Após isto, os cinco conjuntos de cada classe eram combinados em novos cinco conjuntos, de forma a equilibrar a proporção das duas classes em cada conjunto da validação cruzada. Após esta divisão, para validar cada configuração de cada rede, vinte treinamentos eram realizados, selecionando um dos conjuntos para validação, um para teste e deixando o resto para o treinamento. No caso da rede treinada para o terceiro *down*, os vetores de corrida no conjunto de teste foram replicados para equilibrar as duas classes. Para as duas outras redes (primeiro e segundo *down*), como a proporção das duas classes é próxima de 50% e apresenta um número maior de vetores, não foi necessário realizar este tratamento. A média e desvio padrão da taxa de acertos dos vinte treinamentos para cada arquitetura de rede foram utilizados como critério de desempenho. Os resultados para a primeira etapa de treinamento serão vistos na seção seguinte.

O treinamento foi controlado pelo erro de validação, número máximo de épocas e gradiente mínimo. Para a validação, caso não houvesse redução no erro por 200 épocas, o treinamento era interrompido. O gradiente mínimo utilizado foi o padrão do *MATLAB* e o número máximo de épocas para cada treinamento foi configurado para 2000.

Após a seleção do número de neurônios na camada oculta para as três redes, o critério de relevância (Calôba, 1994) foi utilizado para identificar quais os atributos de entrada têm a maior importância para a classificação da rede. Do ponto de vista da classificação, isso permite evidenciar quais são as características do jogo que mais impactam na seleção das jogadas. Quanto ao desempenho da rede, a redução de dimensionalidade reduz o ruído de entrada e a complexidade do modelo. Portanto, o critério foi aplicado com objetivo de reduzir o número de neurônios na camada oculta, aumentar a média da taxa de acertos e reduzir o desvio padrão.

$$R(x_j) = \sum_{i=1}^N \frac{\|out(x_i) - out(x_i|x_{i,j} = < x_j >)\|^2}{N} \quad (1)$$

A relevância é definida da seguinte forma: dado um vetor aplicado na entrada da rede, a relevância da entrada  $x_i$  é definida pelo quadrado da norma da diferença entre a saída medida no modelo treinado e esta saída quando, no vetor de entrada aplicado,  $x_i$  é substituído pela sua média. A relevância é a média de todos esses desvios quadráticos para todos os vetores da base de treino ( $N$  é o número de vetores do conjunto de treino). Portanto, a relevância mede o quanto uma entrada influencia na saída da rede neural. Como os atributos de entrada estão normalizados, substituir a entrada pela média é a mesma coisa que zerar a entrada. Logo, caso a saída da rede não varie quando uma determinada entrada  $x_i$  é zerada, é porque este atributo não é importante para o modelo.

Após o cálculo da relevância, o processo de seleção do número de neurônios na camada oculta foi repetido duas vezes. Na primeira vez considerando apenas as entradas com relevância maior do que 0,1 (após normalizar a relevância, dividindo todas pela maior, para cada rede). Na segunda vez considerando apenas os atributos com relevância maior do que 0,2. Este processo de seleção foi idêntico ao explicado anteriormente, variando o número de neurônios de forma incremental, validando o treinamento com validação cruzada 60-20-20 e utilizando média e variância da taxa de acertos como critério de desempenho. Após esta segunda etapa, os melhores resultados foram comparados, selecionando a melhor rede para cada um dos *downs*. Estes resultados serão apresentados na seção a seguir.

### 3 Resultados

Nesta seção serão apresentados os resultados de cada um dos treinamentos, qual foi a configuração final das redes e o resultado da seleção de atributos.

#### 3.1 Resultados do treinamento

As figuras 1, 2 e 3 apresentam os resultados para a primeira parte do treinamento para o primeiro, segundo e terceiro *down*, respectivamente. A tabela 3 resume os resultados e apresenta o percentual de acertos para a melhor rede dos vinte treinamentos da validação cruzada. NE é o número de neurônios na camada oculta,  $\mu$  é a média da taxa de acertos e  $\sigma$  é o desvio padrão.

Tabela 3: Resultados para a primeira parte dos treinamentos.

Tentativa	NE	$\mu$ (%)	$\sigma$ (%)	Melhor (%)
Primeiro <i>down</i>	20	61,37	5,96	71,43
Segundo <i>down</i>	9	67,46	6,13	78,38
Terceiro <i>down</i>	8	94,41	5,39	97,06

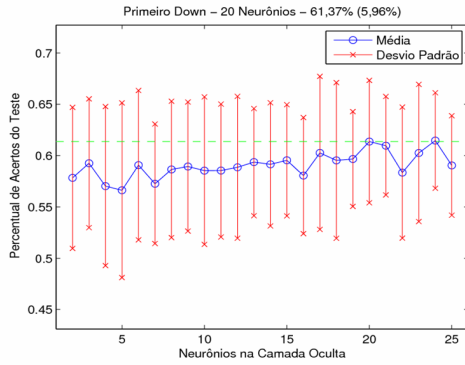


Figura 1: Resultado da primeira etapa de treinamentos para o classificador do primeiro *down*. A melhor configuração encontrada foram 20 neurônios na camada oculta. Taxa de acertos de 61,37% (5,96%).

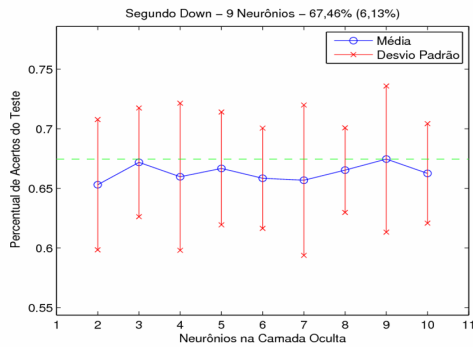


Figura 2: Resultado da primeira etapa de treinamentos para o classificador do segundo *down*. A melhor configuração encontrada foram 9 neurônios na camada oculta. Taxa de acertos de 67,46% (6,13%).

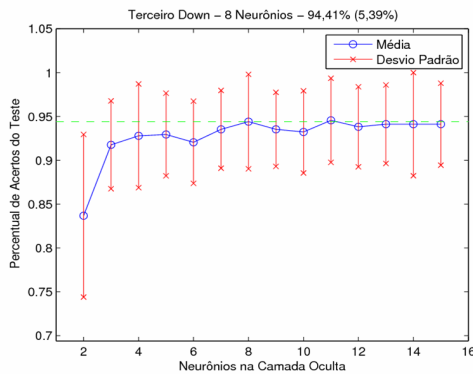


Figura 3: Resultado da primeira etapa de treinamentos para o classificador do terceiro *down*. A melhor configuração encontrada foram 8 neurônios na camada oculta. Taxa de acertos de 94,41% (5,39%).

Após a seleção do número de neurônios na camada oculta, o cálculo da relevância foi efetuado para medir a importância dos atributos. As figuras 4, 5, 6 apresentam o resultado da relevância e a correlação dos atributos com a classe. Isso ilustra como a correlação não seria uma boa medida de relação entre determinado atributo e a classe, para o problema de classificação de jogadas de futebol americano. A tabela 4 apresenta o valor da relevância normalizada para as variáveis.

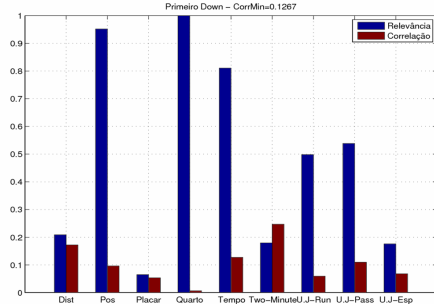


Figura 4: Resultado da relevância e correlação dos atributos para o classificador do primeiro *down*.

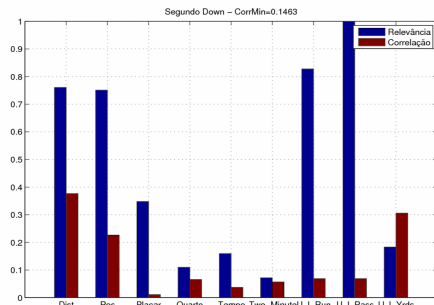


Figura 5: Resultado da relevância e correlação dos atributos para o classificador do segundo *down*.

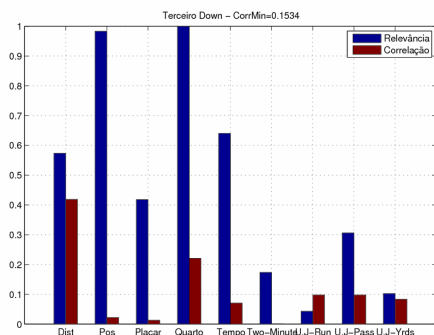


Figura 6: Resultado da relevância e correlação dos atributos para o classificador do terceiro *down*.

Após o cálculo da relevância, dois novos treinamentos foram realizados. O primeiro, eliminando os atributos com relevância abaixo de 0,1.

Tabela 4: Resultados para a análise de relevância.

Atributo	1D	2D	3D
Distância	0,2088	0,7606	0,0052
Posição	0,9517	0,7510	0,0533
Placar	0,0647	0,3477	0,2233
Quarto	1	0,1100	1
Tempo	0,8107	0,1596	0,0999
Two-Minute	0,1796	0,0720	0,2038
U.J-Corrida	0,4984	0,8276	0,5338
U.J-Passe	0,5383	1	0,8053
U.J-Especial	0,1761	-	-
U.J-Jardas	-	0,1833	0,0689

O segundo eliminando os atributos com relevância abaixo de 0,2. Alguns atributos com relevância próxima a 0,2 também foram eliminados neste segundo treinamento. As figuras 7 e 8 apresentam o resultado para o primeiro e o segundo *down*. O resultado para o terceiro *down* não melhorou com a eliminação dos atributos menos relevantes e portanto o primeiro resultado foi mantido.

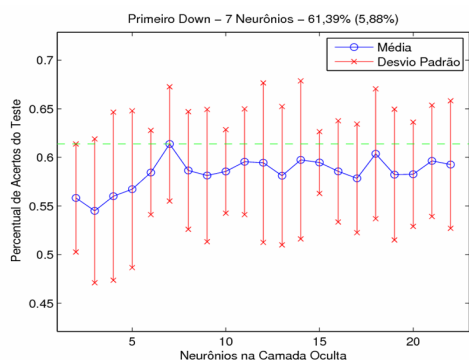


Figura 7: Resultado da segunda etapa de treinamentos para o classificador do primeiro *down*. A melhor configuração encontrada foram 7 neurônios na camada oculta. Taxa de acertos de 61,39% (5,88%). Os atributos eliminados foram: Placar, Distância, *Two-Minute* e U.J. Especial.

O resultado obtido na segunda parte do treinamento melhorou a taxa de acertos ligeiramente para o primeiro *down* e reduziu o número de neurônios na camada oculta de 20 para 7, reduzindo bastante a complexidade do modelo. Para o segundo *down* o número de neurônios na camada oculta se manteve, mas a taxa de acertos média subiu de 67,46% para 72,99%. O melhor resultado para o terceiro *down* foi o do primeiro treinamento. as figuras 9, 10 e 11 apresentam a convergência do treinamento para a melhor taxa de acerto das três redes entre os conjuntos de validação cruzada e a tabela 5 apresenta os resultados finais. NN-1D, NN-2D e NN-3D são as redes treinadas neste trabalho, para o primeiro, segundo e terceiro *down* respectivamente. NN é o resultado obtido em (Rodrigues, 2009). See5 (C4.5) e ID3 foram implementações prévias ao NN utilizando árvores de decisão. A figura 12 apresenta a curva

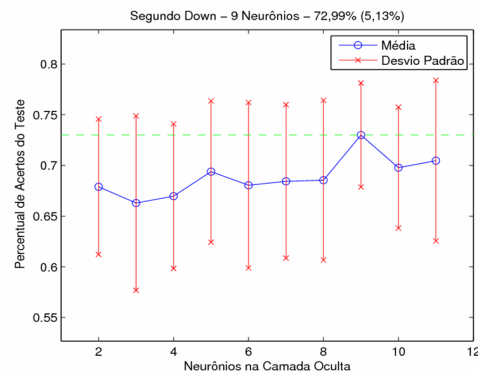


Figura 8: Resultado da primeira etapa de treinamentos para o classificador do segundo *down*. A melhor configuração encontrada foram 9 neurônios na camada oculta. Taxa de acertos de 72,99% (5,13%). Os atributos eliminados foram: *Two-Minute*, Quarto, Tempo e U.J. Jardas.

ROC para cada rede, para as duas classes.

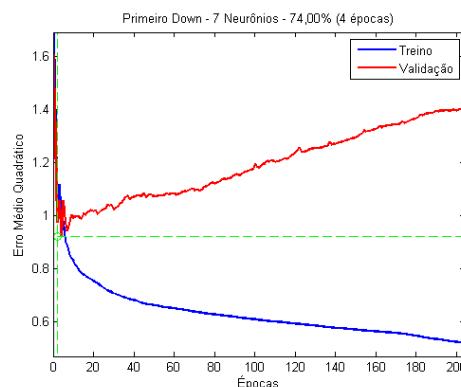


Figura 9: Convergência do treinamento para a melhor rede do primeiro *down*. Encerrada pelo erro de validação, melhor resultado de 74,00% na quarta época.

### 3.2 Discussão

O resultado da rede para o primeiro *down* foi pior dos resultados. No contexto do futebol americano, a escolha da jogada da primeira tentativa é a mais livre das três, já que existirão outras duas tentativas para tentar garantir o avanço. Portanto, o equilíbrio 50-50 para as duas classes possivelmente é proposital para não criar nenhum viés nesta tentativa (e prejudicar as tentativas de detectar algum padrão). A análise de relevância mostra que a posição no campo e o tempo de jogo são os atributos mais importantes para a seleção da jogada. Isto faz bastante sentido para o segundo e o quarto quarto, principalmente, quando o time está jogando “contra o relógio” e deve avançar mais jardas num intervalo menor de tempo, forçando aumentar o jogo de passes.

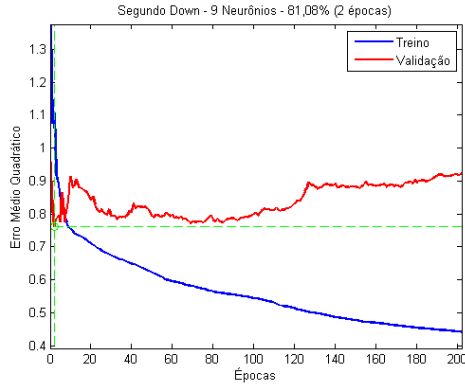


Figura 10: Convergência do treinamento para a melhor rede do segundo *down*. Encerrada pelo erro de validação, melhor resultado de 81,08% na segunda época.

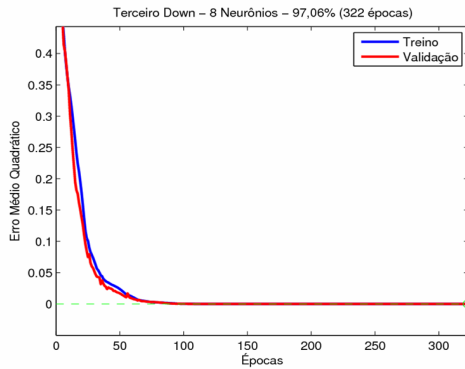


Figura 11: Convergência do treinamento para a melhor rede do terceiro *down*. Encerrada pelo gradiente mínimo, melhor resultado de 97,06% na época número 322.

Tabela 5: Resultados Finais.

Modelo	$N$	$\mu$ (%)	$\sigma$ (%)	Melhor (%)
NN-1D	7	61,39	5,88	74,00
NN-2D	9	72,99	5,13	81,08
NN-3D	8	94,41	5,39	97,06
NN	10	73,00	-	73,00
See5	-	69,00	-	69,00
ID3	-	64,00	-	64,00

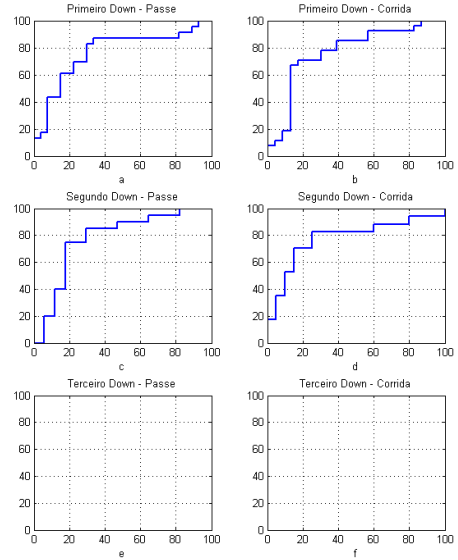


Figura 12: curvas ROC para as redes. A primeira coluna apresenta os resultados para a classe Passe. A segunda coluna para a classe Corrida. Cada linha corresponde a rede treinada para primeiro, segundo e terceiro *down*.

O resultado para o segundo *down* foi comparável a implementação em (Rodrigues, 2009) e embora exista um equilíbrio 50-50 para as duas classes, a taxa de acertos foi maior. Como só existe apenas mais uma tentativa para avançar jardas (a quarta tentativa em geral é utilizada para times especiais), a escolha da jogada nesta tentativa fica mais dependente da primeira jogada. Isto é confirmado pela relevância, que diminuiu completamente o resultado para o tempo de jogo, elevou a relevância da distância para o *first down* e elevou bastante a relevância do que ocorreu na última jogada. A diferença da relevância do primeiro para o segundo *down* é bastante interessante e permite identificar o que seria o comportamento do *quarterback* na escolha das jogadas.

O terceiro *down* obteve o melhor resultado, o que já era esperado já que a maior parte das jogadas desta tentativa são passes. Mesmo assim, a rede neural conseguiu elevar o acerto, ao equilibrar as duas classes no treinamento replicando vetores de corrida. A relevância voltou para a distância, posição e tempo, já que esta é normalmente a última tentativa para garantir o avanço.

Do ponto de vista geral da modelagem do *quarterback* e classificação, os resultados utilizando três redes neurais foi melhor do que apenas uma como em (Rodrigues, 2009), já que elevou o acerto da terceira tentativa e permitiu tirar conclusões mais interessantes sobre a relevância.

### 3.3 Trabalhos futuros

Existem dois pontos principais a melhorar. O aumento da base de dados, para enriquecer a amostra de treinamento, para modelar novos *quarterbacks* e a aquisição de novos atributos que possam ser mais relevantes para a classificação.

### Referências

- Calôba, L.P. Poppe, C. (1994). Identifying relevant features for a real-time discriminating process under high-event rate conditions.
- Haykin, S. (2007). *Redes Neurais - Princípios e Práticas*, 2nd edn, Bookman.
- NFL (2006). N.f.l. play-by-play, <http://www.nfl.com/teams/schedule?team=IND\&season=2006\&seasonType=REG>.
- NFL (2008). N.f.l. beginner's guide to football, <http://www.nfl.com/rulebook/beginnersguidetofootball>.
- Pines, J. Marz, N. (2005). Machine learning in football play-calling, <http://www.stanford.edu/class/cs229/proj2005/PinesMarz-BeatCal.pdf>.
- Quinlan, R. (1986). Induction of decision trees, *Machine Learning, Volume 1, Número 1* pp. 81–106.
- Quinlan, R. (1994). *C4.5 Programs for Machine Learning*, Morgan Kaufmann.
- Riedmiller, M. (1994). Rprop - description and implementation details.
- Rodrigues, D. S. Seixas, J. (2009). Predição de jogadas de futebol americano utilizando redes neurais.