Estimação de Energia via Redes Neurais

DISCIPLINA: FILTROS DIGITAIS

Marcos Oliveira

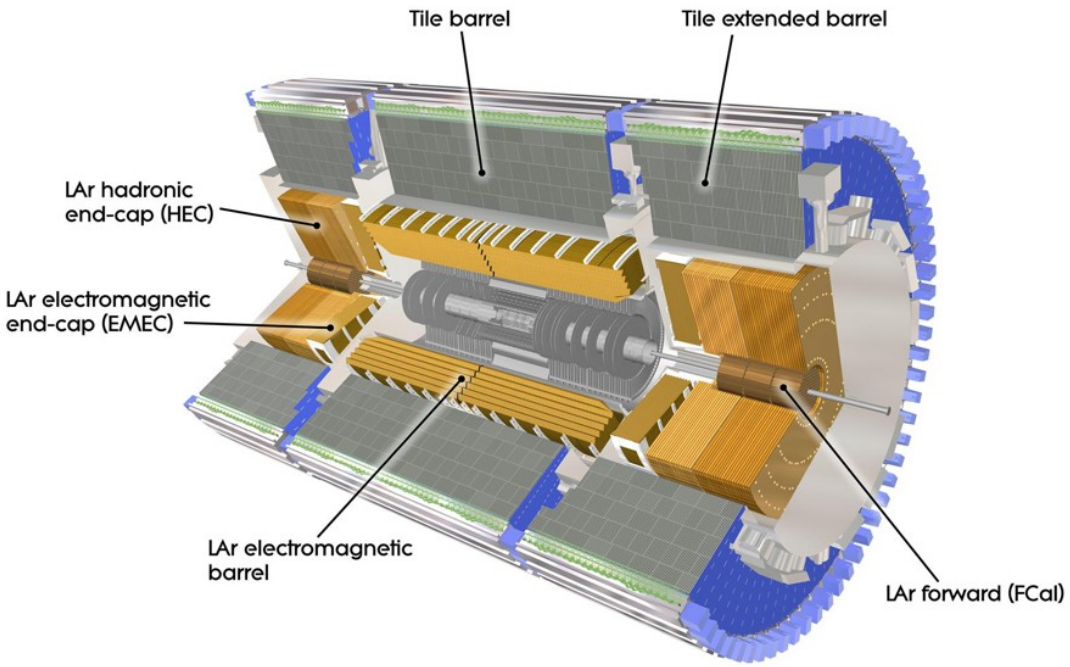
Genebra, 5 de Outubro de 2013.

Sumário

- Calorímetro de telhas do ATLAS
- Contextualização
- Motivação
- Análise
- Implementação
- Projetando uma rede
- Recordando sistemas aritméticos binários
- Avaliação do número de entrada e do número de neurônios
- Avaliações de precisão finita
- Avaliação da saída
- Geração dos arquivos de implementação
- Detalhes do pacote de software criado
- Conclusão

Calorímetro de telhas do ATLAS

- Calorímetro de 10800 canais.
- Divididos em quatro barris (EBC, LBC, LBA, EBA).
- Segmentado no primeiro nível de trigger em 1920 torres.
- Juntamente com os decetores de múon e outros calorímetros compõe importante etapa de seleção de eventos e de análises de física.



Contextualização

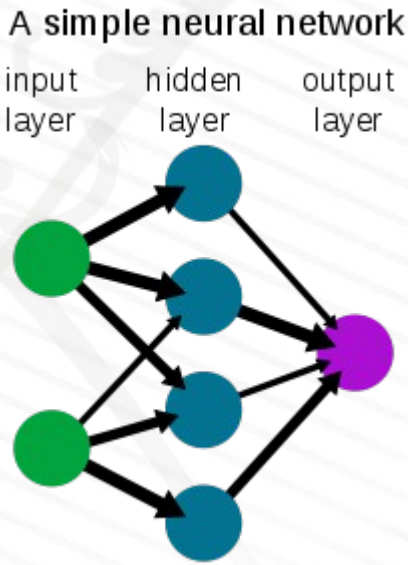
- **Estimação de energia em calorimetria:**
  - ***Seleção de dados (trigger)***
    - Inicialmente é de vital importância a estimação da energia depositada em regiões do detector a fim de decidir se determinado evento é de interesse ou não;
    - Essa estimação geralmente é realizada de forma mais grosseira, como por exemplo, no sub-detector TileCal isto é feita com uma segmentação reduzida a partir da soma de 3 à 6 canais de leitura em apenas um sinal, que chamamos de sinal de uma torre.
  - ***Análises de física (offline analysis)***
    - Os dados de energia são avaliados, caso indiquem a possibilidade de um evento de interesse, o sistema de trigger requisita o armazenamento daquele determinado evento;
    - Neste momento a estimação de energia pode ser feita com maior segmentação, no caso do TileCal isto é realizado individualmente para todos os canais de leitura.

Motivação

- *Atualmente a estimação de energia é realizada a partir de um filtro denominado filtro ótimo, que vem demonstrando ser eficaz para ambientes de baixa taxa de empilhamento de eventos (pile-up).*
- *O grupo de pesquisa do grupo de processamento de sinais da UFJF vem investigando métodos alternativos para a estimação de energia, especialmente para cenários de maior taxa de empilhamento de eventos.*
- Resultados obtidos ao decorrer dos últimos anos indicam que a implementação de um filtro casado apresenta resultados superiores ao filtro ótimo para cenários de alta luminosidade (com pile-up).
- **Neste trabalho propõe-se a investigação da aplicação de Redes Neurais para a estimação de energia, uma vez que o sinal adquirido possui características não lineares, devido a características intrísecas do detector.**

Análise

- Para tal análise será utilizado um banco de dados de simulação de eventos para taxa de empilhamento de eventos variável.
- Desta forma o método proposto será avaliado comparativamente com um método linear, como os filtros de mínimo erro médio quadrático.
- Será avaliado a melhor combinação de número de neurônios e camadas tendo em vista o compromisso de admissível complexidade “computacional”.



Implementação

- A implementação será realizada em hardware conhecido como FPGA (Arranjo de Portas [lógicas] Programável em Campo).
- A implementação de algoritmos de alta complexidade aritmética traz consigo inúmeros fatores que devem ser avaliados como quantização, topologias de maior eficiência, balanceamento de operações e outros mais.
- Esses fatores influenciam decisivamente na eficácia do sistema e sobretudo no custo de implementação.
- Para este trabalho os fatores de custo de implementação não serão intensamente investigados, uma vez que a maior questão esta relacionada a viabilidade ou não deste tipo de técnica para o contexto préviamente citado.

Projetando uma rede

- Circuitos aritméticos devem ser analisados com especial atenção, desde que, diferente de circuitos lógicos convencionais, eles possuem diversas realizações diferentes.
- Cada realização possuirá desempenho diferentes, e uma tarefa crucial é encontrar aquela que atende as especificações desejadas e com o melhor comprometimento entre o custo de implementação e desempenho.
- Deste modo, várias características da rede devem ser investigadas para a aplicação em estudo.
- Existem alguns fatores importantes que devem ser relembrados para a abordagem deste trabalho.

Recordando sistemas aritméticos binários

- O conceito do valor numérico sobrevive a qualquer sistema numérico, mas sua representação possui grande dependência com o sistema numérico adotado.
- Ao decorrer deste trabalho foi adotado o sistema binário tendo seus bits organizados com a notação de complemento de dois, uma vez que é essencial a representação de aritmética com sinal.
- É importante notar também a necessidade de exceder a fronteira dos números inteiros, neste sentido foi adotado a representação de números reais a partir da divisão da palavra binária por sua parte inteira e sua parte fracionária. Com todos estes subsídios foi possível garantir o funcionamento da rede com desempenho de grande sucesso.

0	0	0	1	1	0	0 ,	0	1	0	1	1	0	0
2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}

$$2^3 + 2^2 + 2^{-2} + 2^{-4} + 2^{-5} = 12{,}34375$$

- Para representar a distribuição de bits do exemplo anterior, adotamos a seguinte notação: (8,7) 8 bits para a parte inteira e 7 bits para a parte fracionária.

Recordando sistemas aritméticos binários

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

- Não foi encontrada uma literatura muito clara sobre a análise de precisão finita, deste modo, o que foi adotado neste trabalho foi produzido a partir de um trabalho de engenharia com raciocínio lógico e com o auxílio de intensas simulações computacionais.
- Num caso geral, uma derivação óbvia que foi obtida durantes as análises pode ser descrita da seguinte forma:
  - Uma vez conhecido o número de bits disponível para a representação de um algarismo, deve-se conhecer primeiro o número de bits necessários para representar a parte inteira deste algoritmo com precisão absoluta.
  - Em seguida deve-se alocar o número de bits necessários na parte fracionária para garantir o erro mínimo desejado.
- Desta forma, o número de bits para a parte inteira deve ser constante num caso geral, e deve ser obtido a partir do intervalo possível de representação. Ja a parte fracionária deve ser dependente da precisão necessária e consequentemente com o desempenho desejado.

Recordando sistemas aritméticos binários

- Outras propiedades devem ser lembradas, por exemplo:
  - A == (i1,f1) ; B == (i2,f2).
    - A * B = (i1+i2,f1+f2).
    - Deste modo é conhecido a representação necessária para o algarismo resultante, bem como a nova posição do ponto decimal.
  - Para a soma, podemos usar a seguinte assunção:
    - A + B = (MAX(i1,i2)+1,MAX(f1,f2)).
    - Mas em geral é aplicado algorismos com representação uniforme num somatório.

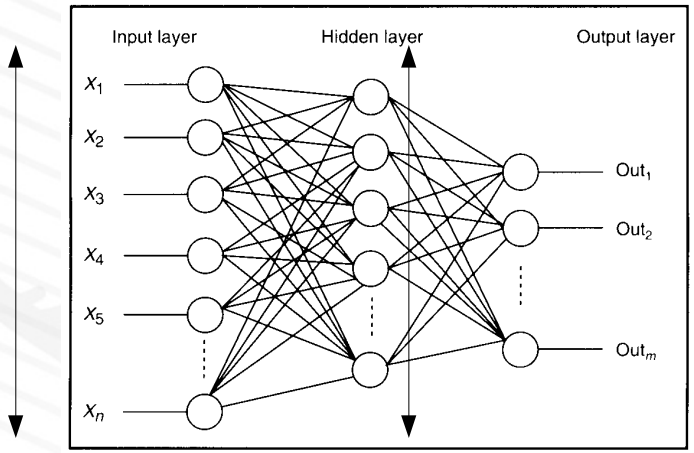
Avaliação do número de entrada e do número de neurônios

- Existe uma dependência entre o número de entrada adotado, o número de neurônios e o desempenho observado.
- Para o contexto de filtros para estimação de energia o número de entradas está associado com o número de amostras utilizadas para a estimação de energia.
- Estes dois parâmetros são de maior importância no desempenho da rede e bem como no seu custo de implementação, deste modo devem ser os primeiros a serem observados.
- Foi testado um conjunto de possíveis combinações de número de entradas e neurônio que teriam coerência com o trabalho proposto. Para cada combinação houve conececutivos processos de treinamento e a rede com o menor erro foi adotada para cada combinação.
- Desta forma foi executado uma iteração de treinamentos variando para número de entradas (L) = {1,3,5,7,9,11} e número de neurônios (Hiddenneurons) = {1,2,3,4,5}.

Avaliação do número de entrada e do número de neurônios

- Vale ressaltar que neste ponto não foi analisado ainda o efeito da precisão finita.

- (L) = {1,3,5,7,9,11}

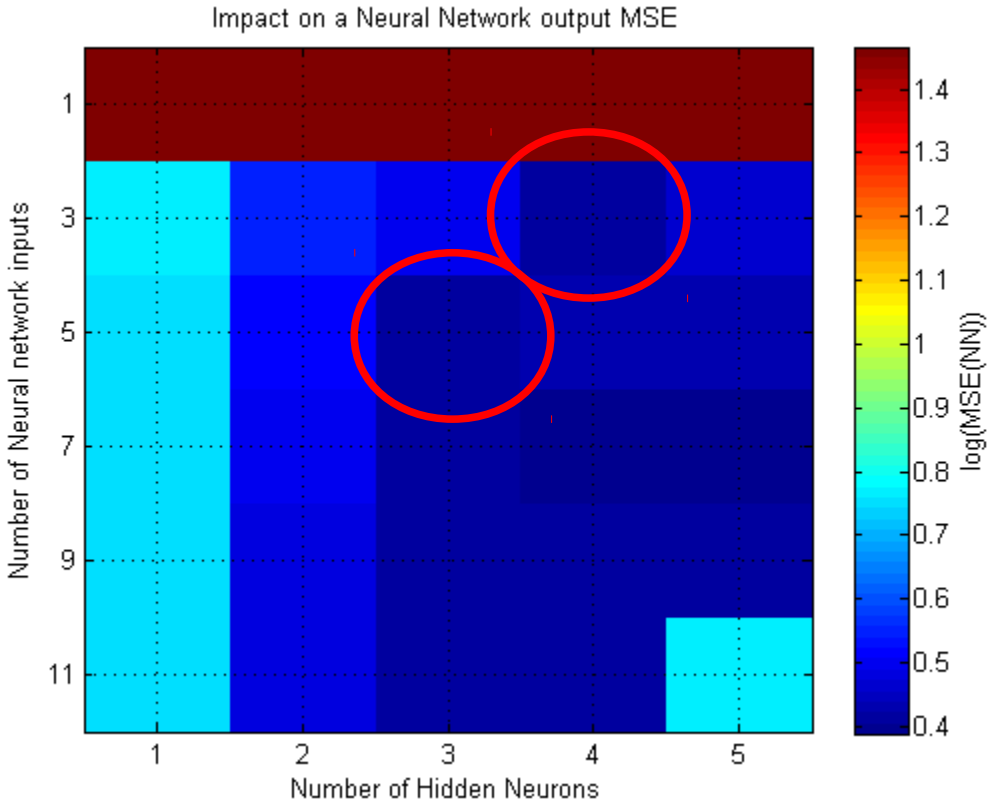


- (Hiddenneurons) = {1,2,3,4,5}.

- Foi utilizado dados de entrada digitalizados e com uma primeira suposição de resolução de 14 bits, como por exemplo pode ser encontrado no circuito ADS5421.

Avaliação do número de entrada e do número de neurônios

- Resultados obtidos:



Pontos de bom comprometimento entre desempenho e complexidade de implementação.

Avaliação do número de entrada e do número de neurônios

- Resultados obtidos:

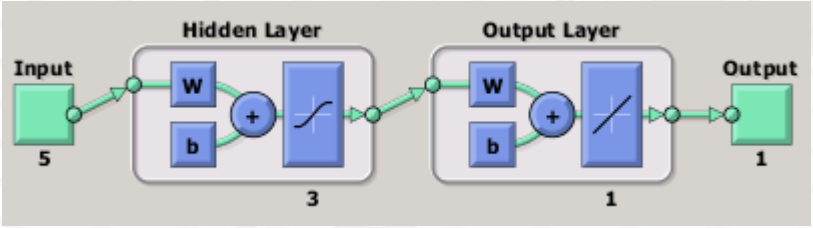
Número de entradas	MSE(LS)
1	75,853
3	46,403
5	41,485
7	39,829
9	39,603
11	39,380

Análise do valor MSE para diferente configurações de rede					
Numero de entradas	Número de neurônios na camada escondida				
	1	2	3	4	5
1	29,277	29,127	29,263	29,266	29,263
3	5,842	3,595	3,096	2,611	2,930
5	5,617	3,312	2,592	2,664	2,671
7	5,572	3,112	2,617	2,446	2,449
9	5,552	3,078	2,595	2,560	2,607
11	5,543	3,039	2,553	2,559	5,812

- Desta forma para esse trabalho foi adotado **L = 5** e **HiddenNeurons = 3**, desde que esta relação possui um bom comprometimento entre custo de implementação e desempenho.

Avaliação do número de entrada e do número de neurônios

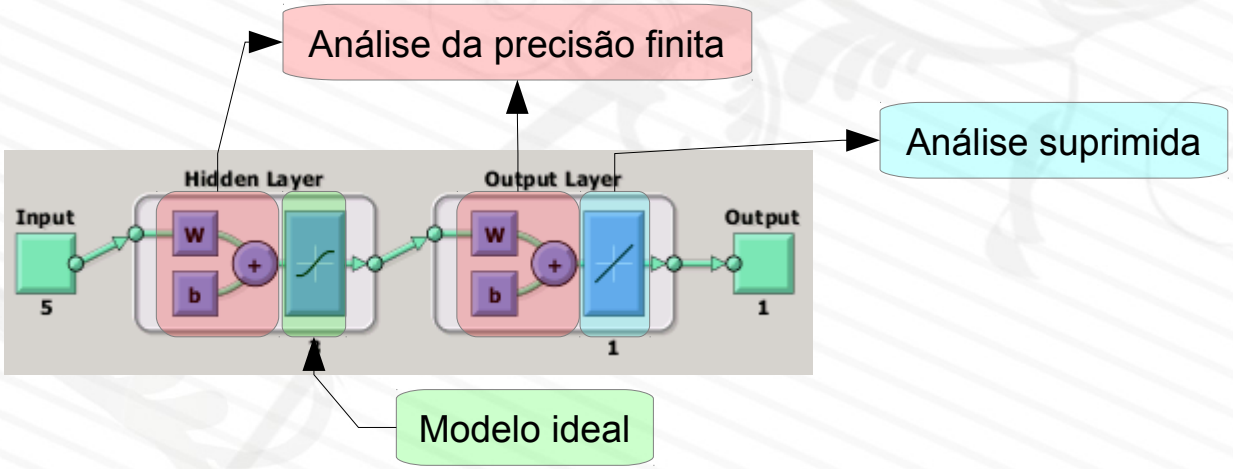
- Uma vez decido os valores de L e Hidden Neurons, a rede adotada pode ser representada pelo seguinte diagrama:



- Isto significa que teremos 5 entradas definidas a partir de registradores em cascata com o dado de entrada conectado ao primeiro registrador.
- Em seguida haverá um produto matricial entre os pesos e a entrada, logo a seguir, uma função de tranferência não linear para cada um dos três neurônios de camada escondida.
- E por último, na camada de saída, haverá um novo produto matricial com os pesos da camada de sáida com os resultados de cada neurônio da camada escondida, logo a seguir uma função de tranferência linear.

Avaliação de precisão finita dos pesos da rede

- Num primeiro momento foi avaliada o efeito da precisão finita somente para os pesos da rede neural, deste modo, as funções de ativação foram modeladas como funções ideais.

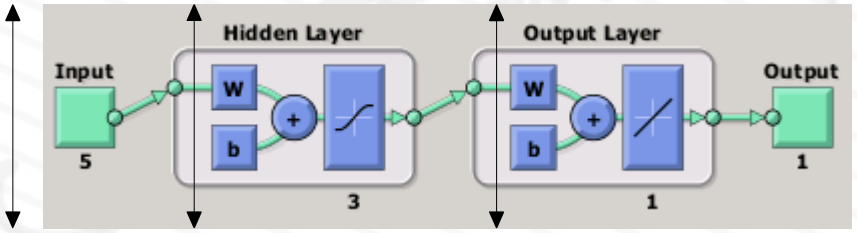


- Neste sentido trabalharemos com a superposição dos efeitos.
- Deste modo, para a rede adotada podemos avaliar o impacto da distribuição do número de bits para pesos de camada escondida e de saída no valor MSE da saída da rede, para funções de ativação ideais.

Avaliação de precisão finita dos pesos da rede

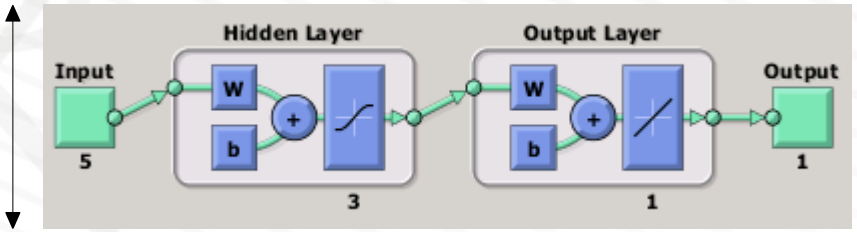
- Então faremos uma iteração variando a representação dos pesos de camada escondida iniciando de um bit até 20 bits, e para cada uma destas iterações será feita uma nova iteração variando a representação dos pesos da camada de saída iniciando novamente em 1 bit e indo até 20 bits.

Realização



14 bits 1 à 20 bits 1 à 20 bits

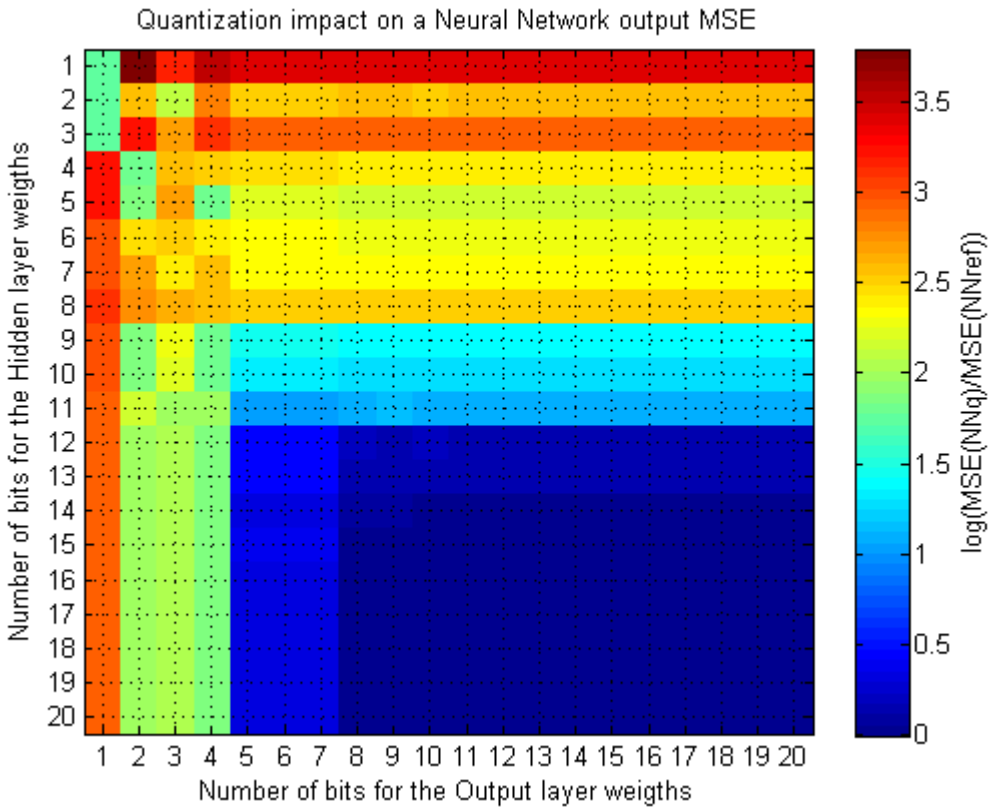
Referência



- Isto siginica que para cada uma das 20 realizações da camada escondida existem mais 20 realizações de camada de saída, resultando em 400 realizações diferentes.

Avaliação de precisão finita dos pesos da rede

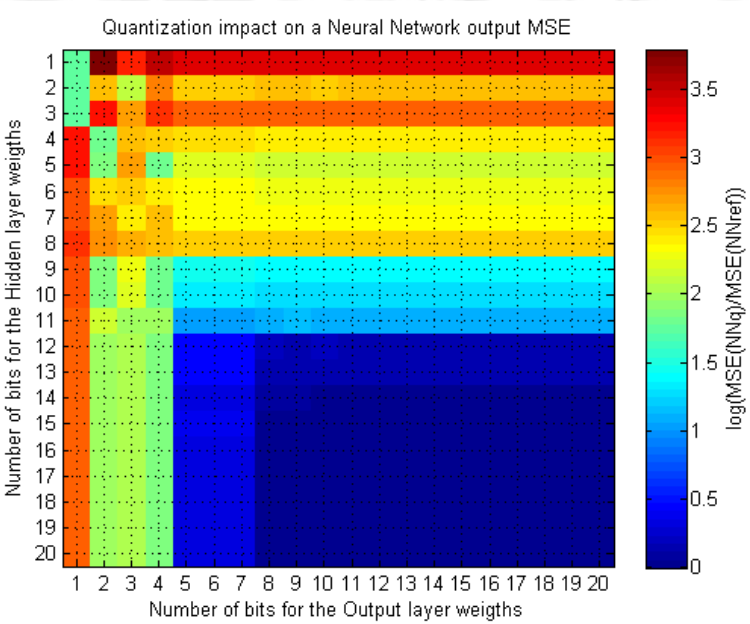
- Esta extensa análise resulta no seguinte histograma:



- Num primeiro momento deve-se observar que foi utilizada uma razão entre a MSE da rede com o efeito da precisão finita e com a MSE da rede ideal. De modo semelhante a uma relação sinal+ruído/sinal, em que o ruído e o erro devido a precisão finita.

Avaliação de precisão finita dos pesos da rede

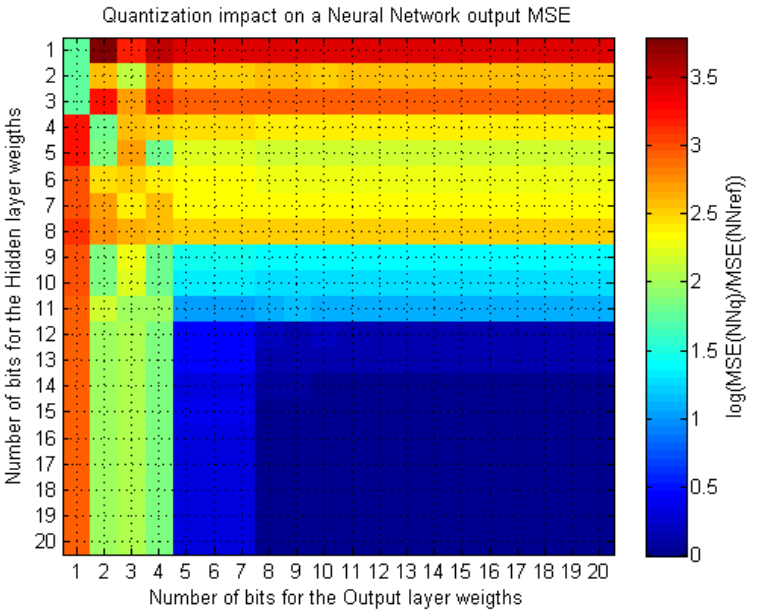
- Num primeiro momento nota-se que o comportamento da razão dos valores MSE não seguem um padrão muito trivial.
- Conforme esperado para baixo número de bits em ambas camadas a relação se torna bem alta, alcançando valores de MSE 3000 vezes maior que o valor de referência.
- Nota-se que mesmo utilizando alta precisão para uma camada e baixa precisão para a outra, o fato de uma delas introduzir o erro impacta severamente o valor MSE de saída da rede.



- Nota-se também que, para a topologia em questão, a representação com qualidade dos pesos da camada escondida se mostram mais determinantes no valor MSE da saída do que os pesos da camada de saída, uma vez que existe a propagação do erro da camada de entrada na camada de saída mais o erro introduzido pelo efeito da precisão finita na camada de saída.

Avaliação de precisão finita dos pesos da rede

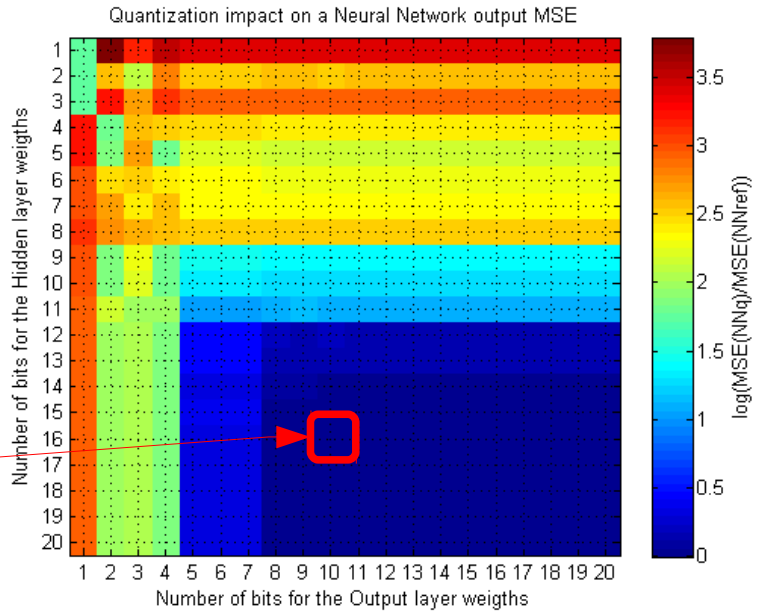
- Neste sentido nota-se que para um representação dos pesos da camada escondida com precisão igual ou maior 14 bits e dos pesos da camada de saída com precisão igual ou maior a 10 bits, alcança-se um relação dos valores de MSE(NNq) < 1.1 MSE(NNref).
- Para este trabalho foi adotado um relação mais restritiva, limitando MSE(NNq) > 1.01 MSE(NNq) o que equivale a limitar o erro médio quadrático em **1%** do valor MSE da rede de referência, o que representa:



$$\log\left(\frac{MSE(NNq)}{MSE(NNref)}\right) < log(1.01) < 0.0043$$

Avaliação de precisão finita dos pesos da rede

- Para tanto, é mais interessante criar uma tabela com os valores da relação para o intervalo de erro menor que 10 %.
- Deste modo concluímos que 16 bits para os pesos da camada escondida e 10 bits para a camada de saída são suficientes para garantir erro < 1%.



Realização com erro < 1 %

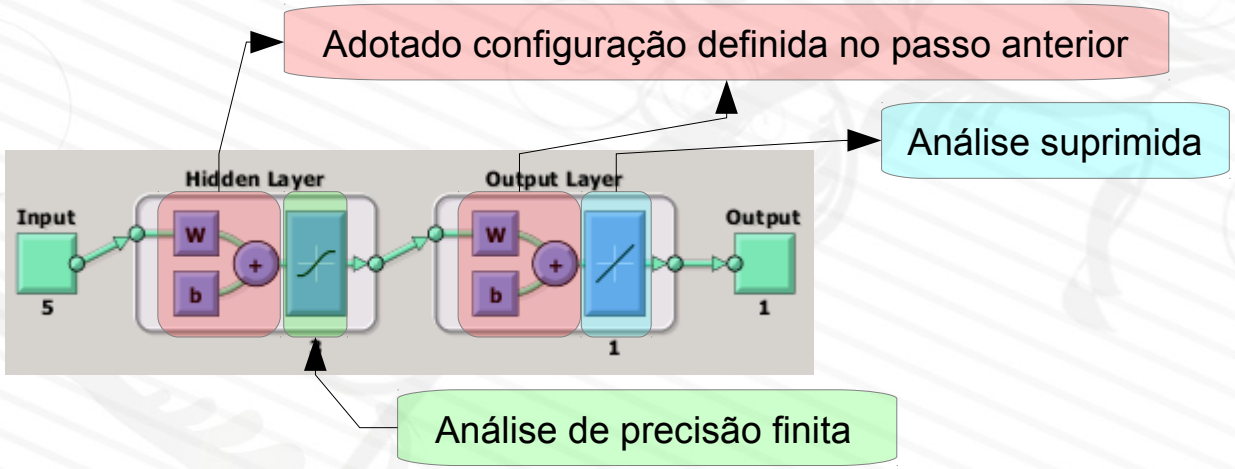
Número de bits para camada escondida	Número de bits para a camada de saída										
	10	11	12	13	14	15	16	17	18	19	20
14	0.0358	0.0380	0.0416	0.0370	0.0382	0.0379	0.0377	0.0380	0.0379	0.0380	0.0379
15	0.0419	0.0313	0.0275	0.0322	0.0305	0.0309	0.0311	0.0308	0.0309	0.0308	0.0309
16	0.0042	0.0007	0.0006	0.0003	0.0001	0.0002	0.0001	0.0001	0.0001	0.0001	0.0001
17	0.0050	0.0052	0.0065	0.0040	0.0045	0.0045	0.0044	0.0045	0.0045	0.0045	0.0045
18	0.0056	0.0011	0.0003	0.0008	0.0004	0.0005	0.0006	0.0005	0.0005	0.0005	0.0005
19	0.0045	0.0007	0.0005	0.0003	0.0001	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
20	0.0042	0.0007	0.0004	0.0002	0.0000	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001

Avaliação de precisão finita dos pesos da rede

- O conjunto de códigos criados neste trabalho deriva uma extensa lista de parâmetros sobre a implementação em hardware da rede neural proposta.
- Entre eles sabemos como será a distribuição entre a parte inteira e a fracional para os pesos propostos.
- Descrição gerada pelo software:
  - Quantization of Input data: Integer part: 9 bits, Fractional part: 5 bits and Total : 14 bits.
  - Quantization of Input layer weights: Integer part: 2 bits, Fractional part: 14 bits and Total : 16 bits.
  - Quantization of Hidden layer weights: Integer part: 9 bits, Fractional part: 1 bits and Total : 10 bits.
- Desta forma podemos derivar o formato do dado após o produto matricial.
  - $Q = (9,5)*(2,14) + (9,5)*(2,14) + (9,5)*(2,14) + (9,5)*(2,14) + (9,5)*(2,14)$
    $= (11,19) + (11,19) + (11,19) + (11,19) + (11,19)$
    $= (12,19) + (12,19) + (11,19)$
    $= (13,19) + (11,19)$
    $= (14,19) =$ **33 bits !**

Avaliação de precisão finita da função de ativação

- Agora neste momento iremos avaliar o impacto da digitalização da função de ativação.



- Deste modo, precisamos avaliar, será mesmo necessário uma LUT de 2^33 palavras?
- Para ter uma idéia, isto seria igual a: 8.589934592×10^9 palavras.

Avaliação de precisão finita da função de ativação

- Agora neste momento iremos avaliar o impacto da digitalização da função de ativação.

- Mesmo que por um instante admitisse-mos utilizar 33 bits para, qual seria a representação para a imagem da função de ativação?

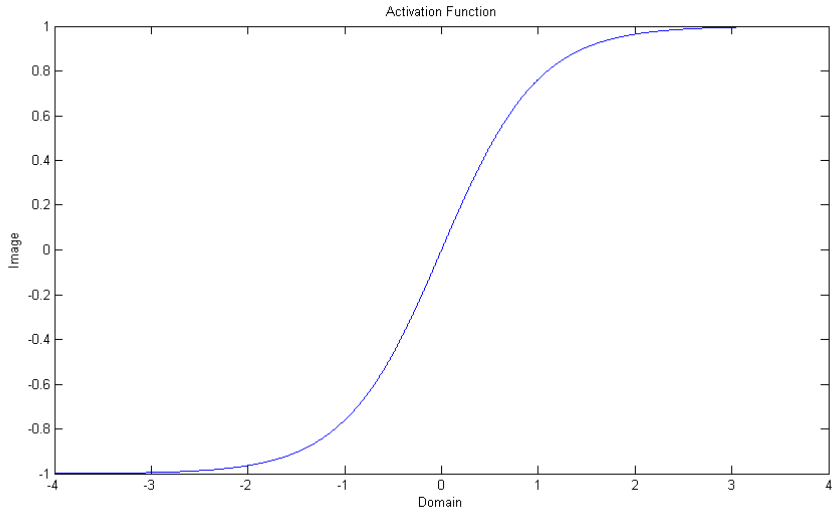
Avaliação de precisão finita da função de ativação

- Podemos avaliar este problema em pequenas partes.

- Mesmo que por um instante admitisse-mos utilizar 33 bits para, qual seria a representação para a imagem da função de ativação?

Avaliação de precisão finita da função de ativação

- É bastante fácil definir a parte inteira, uma vez que a função de ativação possui seu domínio predominate no intervalo [-4, 4].



(3,?) → Função de ativação → (?,?)

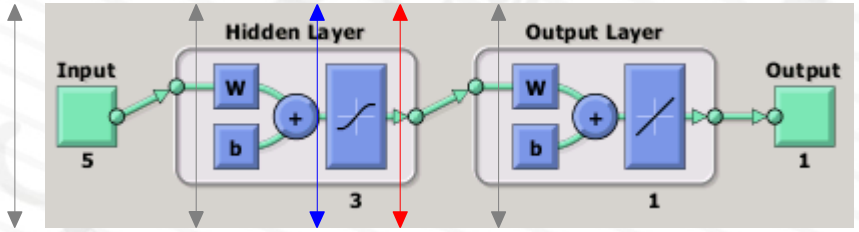
obs: Otimizando a parte inteira para 3 bits resulta num novo intervalo igual à [-4,4-2^{-frac}].

Avaliação de precisão finita da função de ativação

- Para determinarmos a largura as características de quantização da função de ativação podemos proceder de forma semelhante ao que foi feito anteriormente, mas neste momento adotaremos os parâmetros obtidos anteriormente para representar o efeito da precisão finita nos pesos e iterar novos parâmetros para o domínio e imagem da função de ativação.

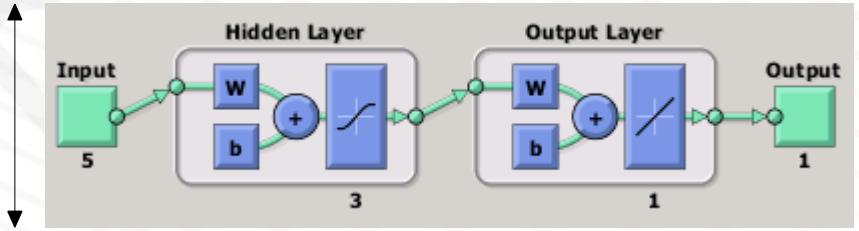
Realização

1 à 20 bits 1 à 20 bits



14 bits 16 bits 10 bits

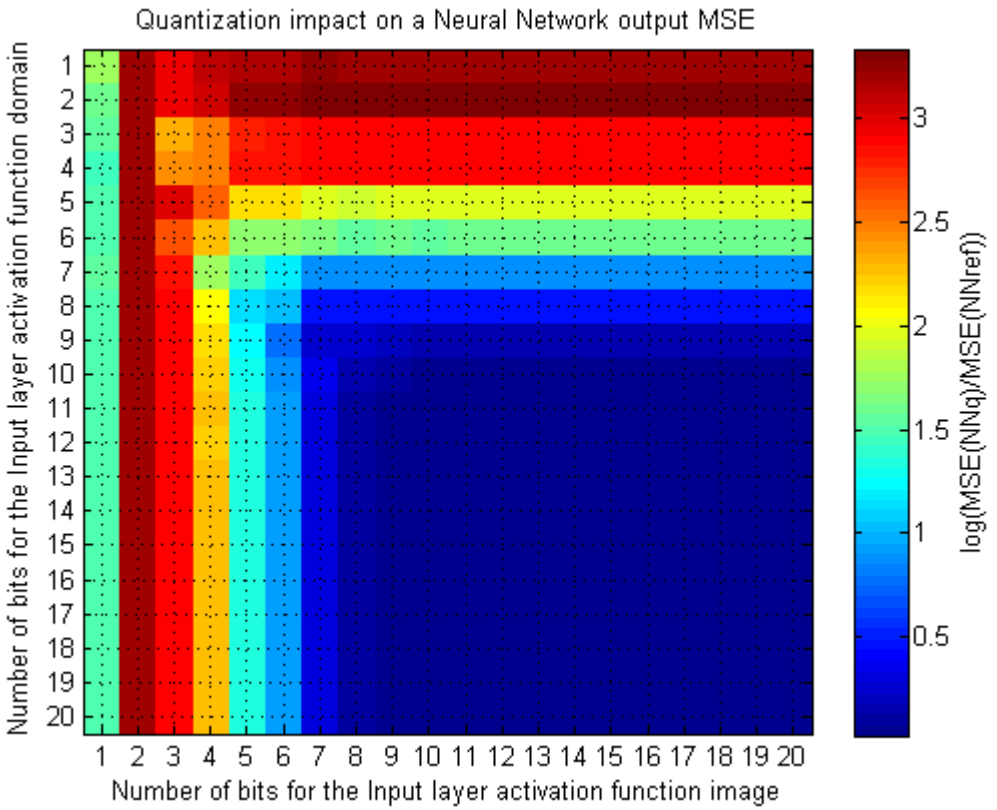
Referência



- Isto siginica que para cada uma das 20 realizações do domínio da função de ativação existem mais 20 realizações da imagem da mesma, resultando em 400 realizações diferentes.

Avaliação de precisão finita da função de ativação

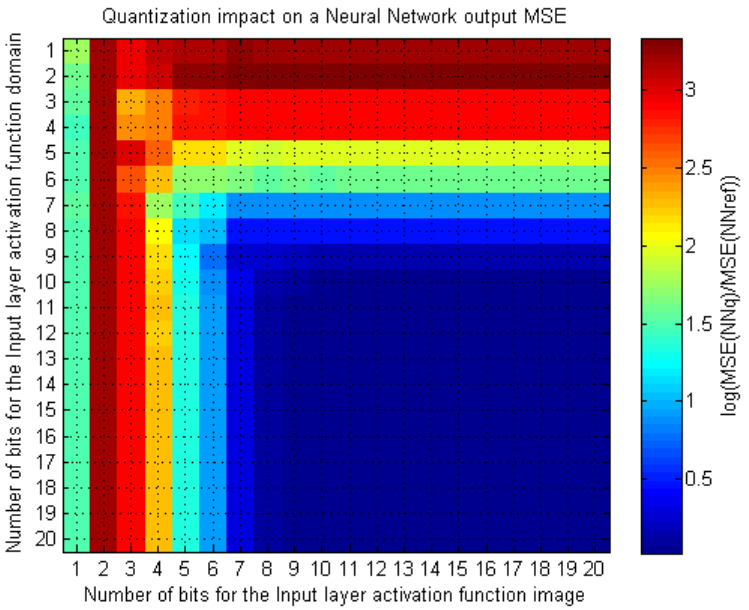
- Novamente esta extensa análise resulta no seguinte histograma:



- De modo semelhante ao anterior, foi utilizada como referência a MSE do sistema original considerando o efeito da precisão finita somente para a entrada. Ativação.
- 16

Avaliação de precisão finita da função de ativação

- Mais uma vez, nota-se que o comportamento da razão dos valores MSE não seguem um padrão muito trivial e semelhante ao visto anteriormente é possível achar valos para MSE de até 1000 vezes maior do que o valor de referência.
- Diferente do histograma anterior, mostra-se que a distribuição de bit no domínio e na imagem resultam em semelhante impacto no desempenho da rede. Isto indica que a as análises indicarão para números semelhantes de alocação de bits para domínio e imagem.



- No entanto ainda é preciso determinar um valor limite para o erro introduzido.
  - De modo semelhante ao obtido anteriormente, obtemos os seguintes valores para a relação de MSE em função da porcentagem da MSE de referência.
  - 1 % = 0.0043; 2% = 0.0086; 3% = 0.0128; 4% = 0.0170; 5% = 0.0212

Avaliação de precisão finita da função de ativação

- A seguir é possível vizualizar a tabela das relações de MSE

- 3 M9K
- 42 LUT, 68 M9K + 4 reg

Número de bits para ao domínio	Número de bits para a imagem da função de ativação										
	10	11	12	13	14	15	16	17	18	19	20
10	0.0592	0.0513	0.0499	0.0484	0.0482	0.0480	0.0478	0.0478	0.0478	0.0478	0.0478
11	0.0289	0.0195	0.0177	0.0162	0.0156	0.0156	0.0154	0.0154	0.0153	0.0153	0.0153
12	0.0203	0.0119	0.0093	0.0079	0.0074	0.0073	0.0072	0.0071	0.0071	0.0071	0.0071
13	0.0155	0.0100	0.0072	0.0060	0.0054	0.0053	0.0051	0.0051	0.0050	0.0050	0.0050
14	0.0161	0.0083	0.0065	0.0053	0.0048	0.0046	0.0045	0.0044	0.0044	0.0044	0.0044
15	0.0169	0.0087	0.0058	0.0053	0.0047	0.0045	0.0044	0.0043	0.0043	0.0043	0.0043
16	0.0169	0.0089	0.0060	0.0050	0.0047	0.0045	0.0044	0.0043	0.0043	0.0043	0.0043
17	0.0167	0.0089	0.0061	0.0051	0.0046	0.0045	0.0044	0.0043	0.0043	0.0043	0.0043
18	0.0168	0.0090	0.0062	0.0051	0.0047	0.0044	0.0044	0.0043	0.0043	0.0043	0.0043
19	0.0167	0.0090	0.0062	0.0052	0.0047	0.0045	0.0043	0.0043	0.0043	0.0043	0.0043
20	0.0167	0.0090	0.0062	0.0052	0.0047	0.0045	0.0043	0.0043	0.0043	0.0043	0.0043

- 5 M9K
- 10 M9K
- 6 M9K
- 7 M9K

- No entanto ainda é preciso determinar um valor limite para o erro introduzido.
  - De modo semelhante ao obtido anteriormente, obtemos os seguintes valores para a relação de MSE em função da porcentagem da MSE de referência.
  - 1 % = 0.0043; 2% = 0.0086; 3% = 0.0128; 4% = 0.0170; 5% = 0.0212

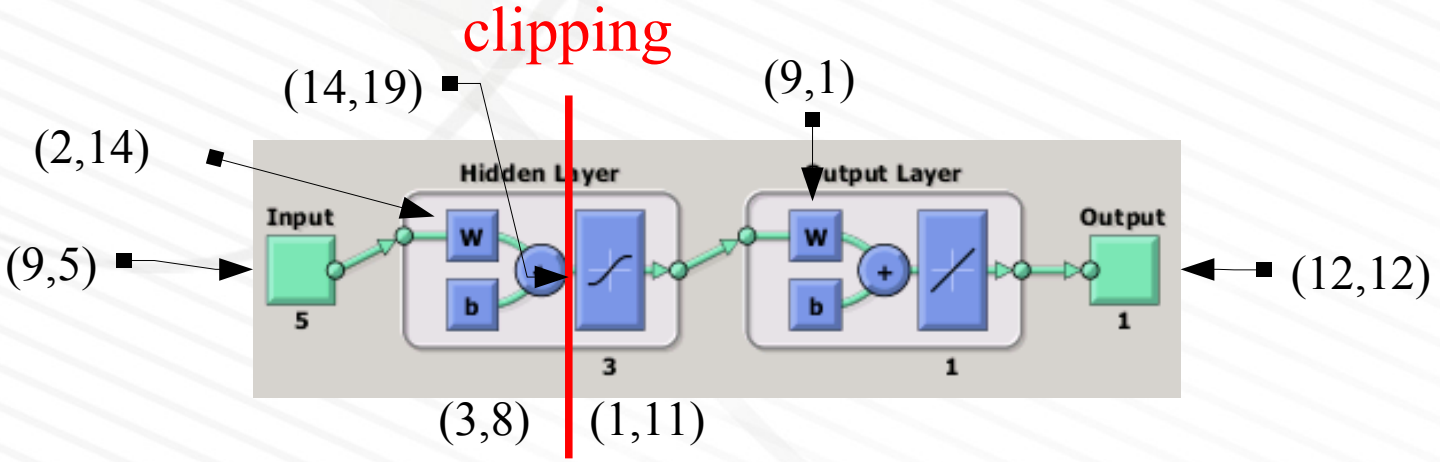
Avaliação de precisão finita da função de ativação

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

- Deste modo para aplicações de baixo custo de implementação a configuração com 11 bits para a representação do domínio e 12 bits para a imagem oferecem uma ótima relação custo benefício, com erro total = 4,16 % em relação à rede neural de referência.
- Adotando estes valores obtemos as seguintes respostas da aplicação de simulação:
  - Quantization of Input layer sum: Integer part: 3 bits, Fractional part: 8 bits and Total : 11 bits.
  - Quantization of Input layer activation function: Integer part: 1 bits, Fractional part: 11 bits and Total : 12 bits.
- Isto significa que o domínio da função de ativação foi reduzido de **33 bits para 11 bits**, o que neste caso significa mover da região de inviabilidade de implementação para a região de implementação de baixo custo.

Avaliação da saída

- Desta forma se a imagem da função de ativação ssuindo 12 bits com a configuração (1,11), podemos derivar a saída da rede da seguinte forma:
  - Y = (1,11)*(9,1) + (1,11)*(9,1) + (1,11)*(9,1)
    = (10,12) + (10,12) + (10,12)
    = (11,12) + (10,12)
    = (12,12)
- Indicando que a saída da rede terá 24 bits com a metade dos mesmos para a parte inteira e a outra metade para a parte fracionária.



Avaliação da saída

- Estimações finais:
  - Quantization of Input layer weights: Integer part: 2 bits, Fractional part: 14 bits and Total : 16 bits.
  - Quantization of Input layer sum: Integer part: 3 bits, Fractional part: 8 bits and Total : 11 bits.
  - Quantization of Input layer activation function: Integer part: 1 bits, Fractional part: 11 bits and Total : 12 bits.
  - Quantization of Hidden layer weights: Integer part: 9 bits, Fractional part: 1 bits and Total : 10 bits.
  - MSE para diferentes métodos
  - MSE referência: 148.8518
  - MSE LS : 39.8614
  - MSE NN : 2.6391
  - MSE NN_y : 2.7489

Geração dos arquivos de implementação

- O pacote proposto neste trabalho também é capaz de gerar os arquivos que representarão os pesos da camada escondida, as LUTs que representarão a função de ativação e os pesos da camada de saída para qualquer topologia e parâmetros definidos ao decorrer desta apresentação.
  - InputLayerWeights.dat
  - InputLayerActiveFuncLUT.dat
  - HiddenLayerWeights.dat

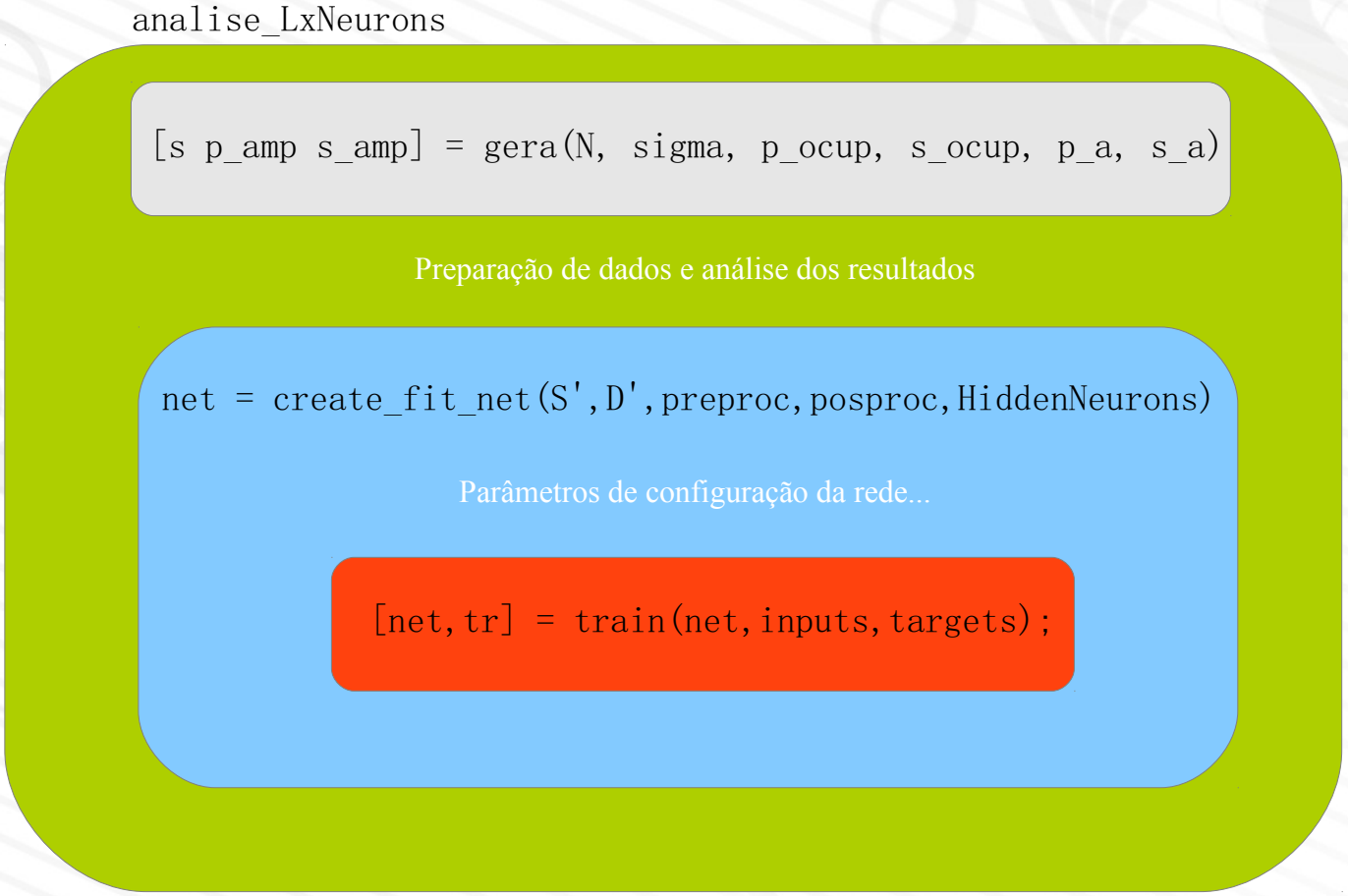
Exemplo:

```
1  0x00000c10
2  0x0000ffeb
3  0x00000080
4  0x0000ff05
5  0x0000008c
6  0x0000ffe2
7  0x0000bba5
8  0x000000b4
9  0x0000fb4a
10 0x00000946
11 0x0000facc
12 0x00000110
13 0x0000254d
14 0x0000004e
15 0x0000fe2f
16 0x00000391
17 0x0000fe02
18 0x00000071
```

- Desta forma todos estes dados de implementação podem ser carregados através de um barramento VME ou a partir de uma métodologia mais moderna, como configuração via Ethernet usando um framework, por exemplo, IPBus.

Detalhes do pacote de software criado

- Software utilizado par a análise de número de entradas em relação ao número de neurônios



Conclusão

- Conforme discutido no trabalho, esta análise é capaz de conduzir a implementação da região de inviabilidade de implementação para a zona de baixo custo de implementação.
- Mostra que é possível obter resultados satisfatórios com rede neurais otimizadas para baixo custo de implementação e chegando alcançar desempenho 20 vezes melhor que o método LS.