

# SCC0270/SCC5809 - Redes Neurais

## Aula 3 - *Multi-Layer Perceptron* (MLP)

Profa. Dra. Roseli Aparecida Francelin Romero  
SCC - ICMC - USP

2018

# Sumário

## 1 Introdução

- Modelo de rede MLP

## 2 Treinamento de redes MLP

- O algoritmo Backpropagation
- Processo de aprendizado
- Exemplo

#### 4 Considerações práticas

- Velocidade de aprendizado
- Termo Momentum
- Modos de treinamento
- Critério de parada
- Generalização
- Normalização
- Inicialização

# Perceptron multicamadas

- Redes de apenas uma camada representam somente funções linearmente separáveis.
- Redes de múltiplas camadas solucionam essa restrição.
- O desenvolvimento do algoritmo *backpropagation* foi um dos motivos para o ressurgimento da área de redes neurais [Rumelhart *et. al*, 1986].

# Sumário

- 1 Introdução
  - Modelo de rede MLP
- 2 Treinamento de redes MLP
  - O algoritmo Backpropagation
  - Processo de aprendizado
  - Exemplo
- 3 Teorema da Aproximação Universal
- 4 Considerações práticas
  - Velocidade de aprendizado
  - Termo Momentum
  - Modos de treinamento
  - Critério de parada
  - Generalização
  - Normalização
  - Inicialização

# Modelo de rede neural com múltiplas camadas.

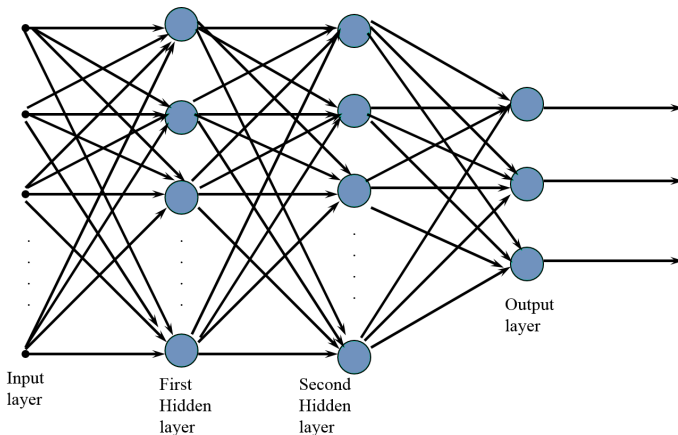


Figura 1: Rede neural *feed-forward* com múltiplas camadas.

# Sumário

- 1 Introdução
  - Modelo de rede MLP
- 2 **Treinamento de redes MLP**
  - O algoritmo Backpropagation
  - Processo de aprendizado
  - Exemplo
- 3 Teorema da Aproximação Universal
- 4 Considerações práticas
  - Velocidade de aprendizado
  - Termo Momentum
  - Modos de treinamento
  - Critério de parada
  - Generalização
  - Normalização
  - Inicialização

## MATERIALS

[illegible]

- O algoritmo Backpropagation

- **Stress** is a response to a stimulus that is perceived as a threat to well-being

### Model 1: The Effect of the Number of Firms

- Velocidade de aprendizado

- 11 2

- 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 104

- Vetor entrada  $\rightarrow$   $\begin{matrix} \text{vetor saída} \\ \text{vetor desejado} \end{matrix} \begin{matrix} - \\ + \end{matrix} ? \rightarrow$   $\begin{matrix} \text{aprendizado} \\ \text{ocorreu} \end{matrix}$



# Aprendizado da rede

- Caso contrário, os pesos são modificados para minimizar o erro:

$$E(w) = \sum_{p=1}^N E_p(w)$$

onde  $N$  é o no. total de padrões e  $E_p$  é o erro quadrático referente a cada par  $p$  apresentado à rede, sendo dado por:

$$E_p = \frac{1}{2} \sum_j (t_{pj} - y_{pj})^2$$

onde:

- $t_{pj}$ :  $j$ -ésima componente do vetor saída desejada.
- $y_{pj}$ :  $j$ -ésima componente do vetor obtido pela rede.

$$w_{ji}(k+1) = w_{ji}(k) - \eta \frac{\partial E_p(w)}{\partial w_{ji}} \Big|_{w(k)}$$

Onde  $\eta$  é uma constante positiva (velocidade de aprendizado).

- Calculando a derivada parcial do  $E_p$ , tem-se:

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial y_{pj}} \cdot \frac{\partial y_{pj}}{\partial v_{pj}} \cdot \frac{\partial v_{pj}}{\partial w_{ji}}$$

- Para se calcular  $\frac{\partial E_p}{\partial y_{pi}}$ , dois casos devem ser considerados:

## Aprendizado da rede

- ① Neurônio  $j$  está na camada de saída.

$$\frac{\partial E_p}{\partial y_{pj}} = -(t_{pj} - y_{pj})$$

$$\therefore \frac{\partial E_p}{\partial w_{ji}} = \underbrace{-(t_{pj} - y_{pj})}_{\delta_{pj}} \cdot \overbrace{y_{pj}(1 - y_{pj})}^{\frac{\partial y_{pj}}{\partial v_{pj}}} \cdot y_{pi}$$

$\frac{\partial E_p}{\partial w_{ji}} = -\delta_{pj} \cdot y_{pi} \rightarrow \text{erro na camada de saída}$

onde  $-\delta_{pj} = \frac{\partial E_p}{\partial v_{pj}}$

# Aprendizado da rede

- ② Neurônio  $j$  está na camada oculta (escondida).
- Nesse caso, não se conhece a expressão do erro.
  - Para obtermos  $\frac{\partial E_p}{\partial y_{pj}}$ , usamos mais uma vez a **regra da cadeia**.

$$\begin{aligned}\frac{\partial E_p}{\partial y_{pj}} &= \sum_k \frac{\partial E_p}{\partial v_{pk}} \cdot \frac{\partial v_{pk}}{\partial y_{pj}} = \sum_k \frac{\partial E_p}{\partial v_{pk}} \cdot \frac{\partial \left( \sum_j w_{kj} y_{pj} \right)}{\partial y_{pj}} \\ &= \sum_k \frac{\partial E_p}{\partial v_{pk}} \cdot w_{kj} = \sum_k (-\delta_{pk} \cdot w_{kj})\end{aligned}$$

$$\therefore \frac{\partial E_p}{\partial w_{ji}} = \left( \sum_k (-\delta_{pk} w_{kj}) \right) \cdot y_{pj}(1 - y_{pj}) \cdot y_{pi}$$

erro na camada oculta

## Aprendizado da rede

- **Observação:** os erros são computados no sentido *backward*. O erro foi chamado de *back-propagado* → algoritmo de aprendizado **backpropagation** (BP).

# Algoritmo *Backpropagation*

- **Inicialização:** pesos iniciados com valores aleatórios e pequenos ( $[-1, +1]$ ).
- **Treinamento - Repita:**
  - Considere um novo padrão de entrada  $x_i$  e seu respectivo vetor de saída  $t_i$  desejado do conjunto de treinamento.
  - **Repita:**
    - Apresentar o par  $(x_i, t_i)$ . **(modo padrão)**
    - Calcular as saídas dos processadores, começando da primeira camada escondida até a camada de saída.
    - Calcular o erro na camada de saída.
    - Atualizar os pesos de cada processador, começando pela camada de saída, até a camada de entrada.
  - **Até que o erro quadrático médio para esse padrão seja  $\leq tol/1$ .**
- **Até que o erro quadrático médio seja  $\leq tol/2$  para todos os padrões do conjunto de treinamento.**

## MATERIALS

© 2011 Blackwell Publishing Ltd

- O algoritmo Backpropagation

- **Stress** is a response to a stimulus that is perceived as a threat or challenge.

### Model 1: The Effect of the Number of Firms

- Velocidade de aprendizado

- Al... 1... 2

- 1 1 1 1 2

# Processo de aprendizado

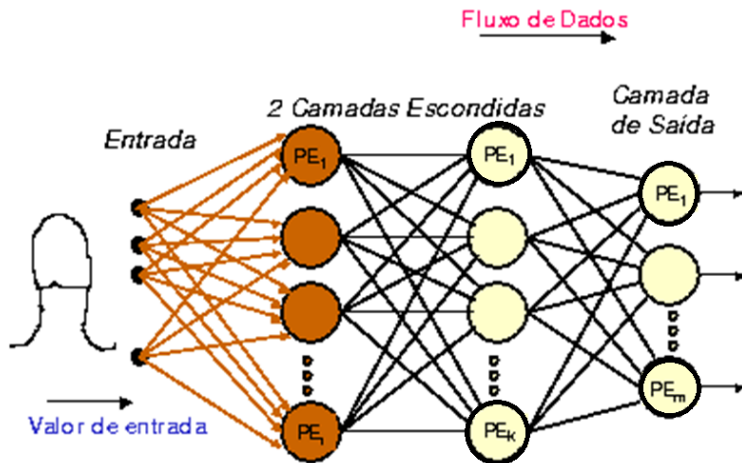


Figura 2: *Feed-forward* (fase 1), primeira camada escondida.



# Processo de aprendizado

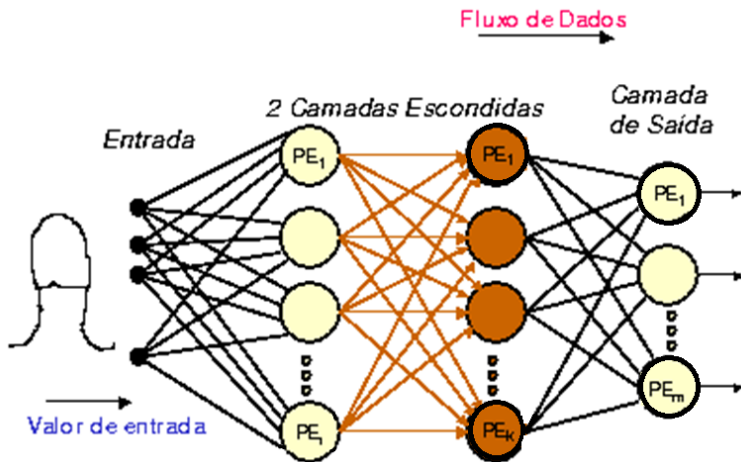


Figura 3: *Feed-forward* (fase 1), segunda camada escondida.

## Processo de aprendizado

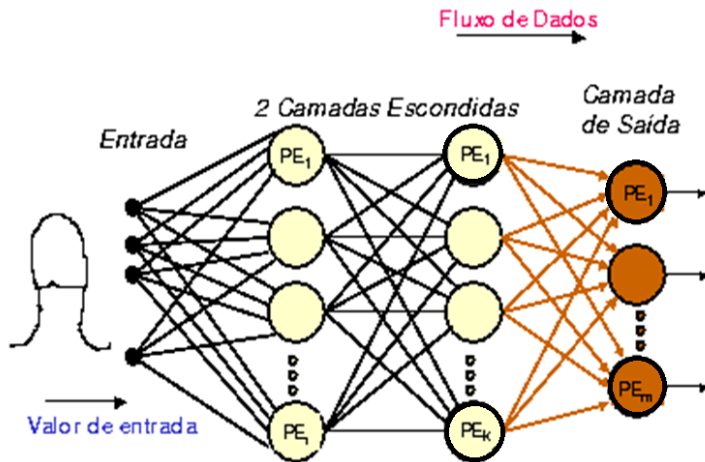


Figura 4: *Feed-forward* (fase 1), camada de saída.

# Processo de aprendizado

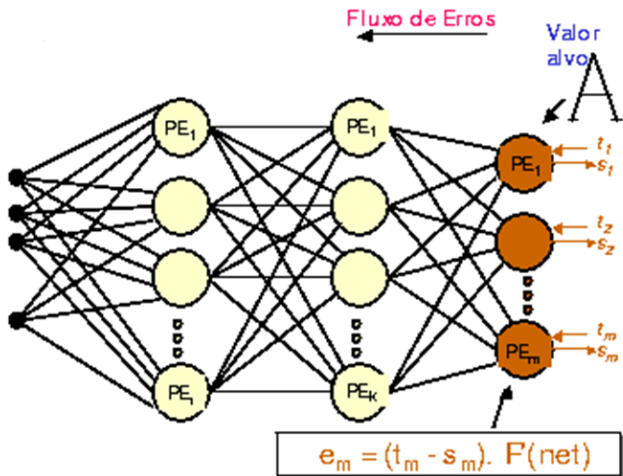
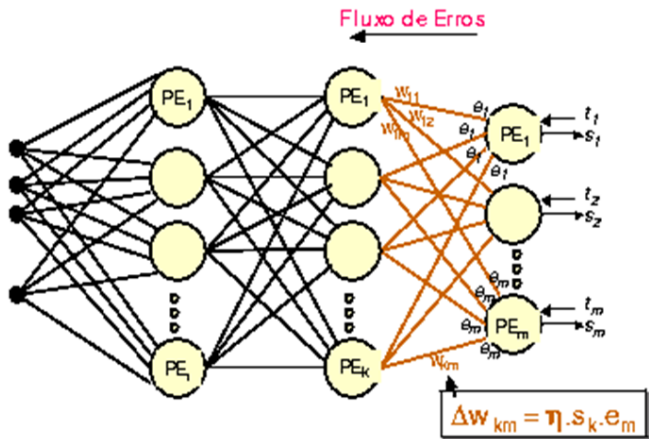


Figura 5: *Feed-backward* (fase 2), cálculo do erro da camada de saída.

# Processo de aprendizado



**Figura 6:** *Feed-backward* (fase 2), atualização dos pesos da camada de saída.

# Processo de aprendizado

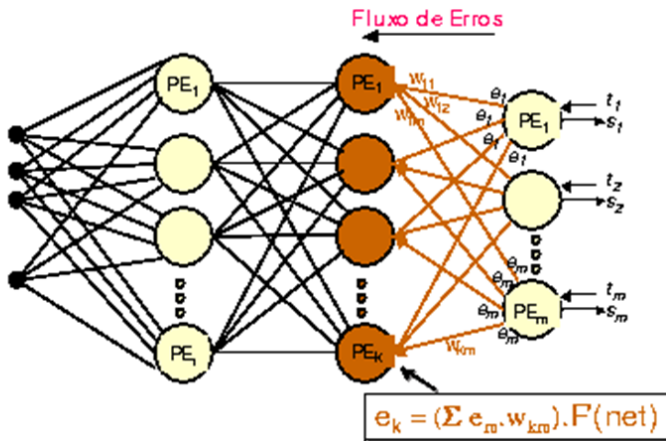
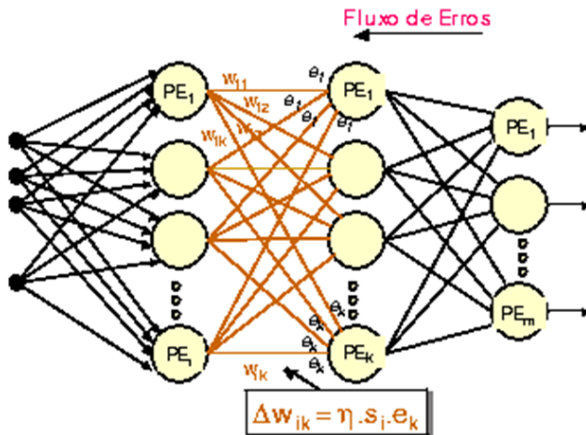


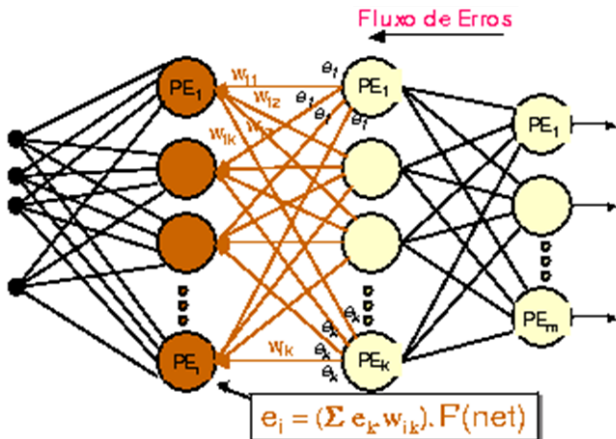
Figura 7: *Feed-backward* (fase 2), cálculo do erro da segunda camada escondida.

# Processo de aprendizado



**Figura 8:** *Feed-backward* (fase 2), atualização dos pesos da segunda camada escondida.

# Processo de aprendizado



**Figura 9:** *Feed-backward* (fase 2), cálculo do erro da primeira camada escondida.

# Processo de aprendizado

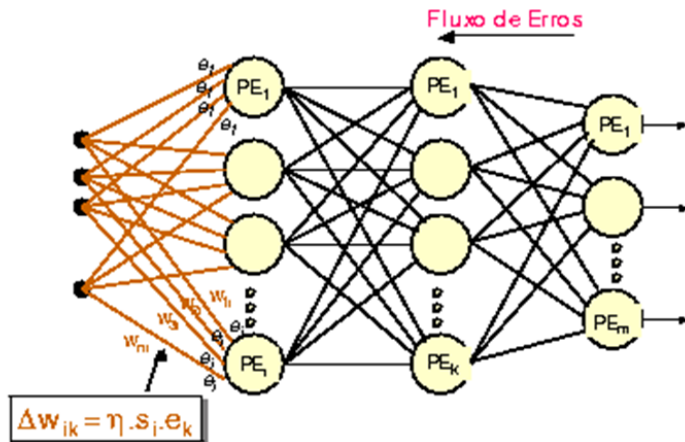


Figura 10: *Feed-backward* (fase 2), atualização dos pesos da primeira camada escondida.



# Processo de aprendizado

- Este procedimento de aprendizado é repetido diversas vezes, até que, **para todos processadores de camada de saída e para todos padrões de treinamento**, o erro seja menor do que o especificado.



# Exemplo - XOR

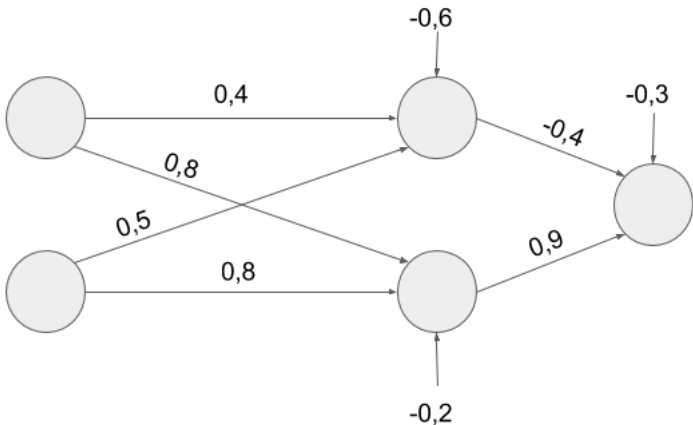


Figura 11: Rede neural inicial. Atualizar os pesos.

### Exemplo

## Exemplo - XOR

Taxa aprendido	0,5					
	t	0	1	2	3	4
Entrada	x.1	1	0	0	1	
	x.2	1	0	1	0	
Saída desejada	y	0	0	1	1	
Pesos	w1_h1	-0,6	-0,5934	-0,5876	-0,5951	-0,6018
	w11_h1	0,4	0,4066	0,4066	0,4066	0,4000
	w21_h1	0,5	0,5066	0,5066	0,4991	0,4991
	w2_h1	-0,2	-0,2097	-0,2217	-0,2092	-0,1968
	w12_h1	0,8	0,7903	0,7903	0,7903	0,8027
	w22_h1	0,8	0,7903	0,7903	0,8028	0,8028
	w1_o	-0,3	-0,3679	-0,4255	-0,3594	-0,2969
	w11_o	-0,4	-0,4390	-0,4595	-0,4278	-0,3995
	w21_o	0,9	0,8456	0,8197	0,8619	0,9020
Camada $h_1$	v1.h1(x)	0,3	-0,5934	-0,0809	-0,1885	
	v2.h1(x)	1,4	-0,2097	0,5686	0,5811	
	f[v1.h1(x)]	0,5744	0,3559	0,4798	0,4530	
	f[v2.h1(x)]	0,8022	0,4478	0,6384	0,6413	
Camada o (saída)	v1.o(h1)	0,1922	-0,1455	-0,1226	-0,0005	
	y' = f[v1.o(h1)]	0,5479	0,4637	0,4694	0,4999	

# Sumário

- 1 Introdução
  - Modelo de rede MLP
- 2 Treinamento de redes MLP
  - O algoritmo Backpropagation
  - Processo de aprendizado
  - Exemplo
- 3 Teorema da Aproximação Universal
- 4 Considerações práticas
  - Velocidade de aprendizado
  - Termo Momentum
  - Modos de treinamento
  - Critério de parada
  - Generalização
  - Normalização
  - Inicialização

# Teorema da Aproximação Universal (TAU)

- Qual é o número mínimo de camadas em uma MLP que fornece uma aproximação para qualquer mapeamento contínuo?
- Cybenko [1989] mostrou pela primeira vez que uma rede com **uma única camada intermediária** é suficiente para aproximar uniformemente qualquer função contínua definida em um hipercubo unitário.

# Teorema da Aproximação Universal (TAU)

- Teorema:** seja  $g(\cdot)$  uma função contínua limitada estritamente crescente. Seja  $I_p$  um hipercubo unitário  $p$ -dimensional e  $C(I_p)$  o espaço das funções contínuas em  $I_p$ . Então, dada qualquer função  $f \in C(I_p)$  e  $\epsilon > 0$ , existe um inteiro  $M$  e constantes reais  $\alpha_i$ ,  $\theta_i$  e  $w_{ji}$ , onde  $i = 1, 2, \dots, M$  e  $j = 1, 2, \dots, p$ , tal que se pode definir:

$$F(x_1, \dots, x_p) = \sum \alpha_i g \left( \sum w_{ji} x_j - \theta_i \right) \quad (1)$$

com

$$|F(x_1, \dots, x_p) - f(x_1, \dots, x_p)| < \epsilon \quad \{x_1, \dots, x_p\} \in I_p$$

# Teorema da Aproximação Universal (TAU)

- As funções *sigmoid* ou logística são contínuas, estritamente crescentes e limitadas, portanto satisfazem as condições impostas para a função  $g(\cdot)$ .
- A equação (1) representa a saída da MLP.
  - A rede tem  $p$  nós de entrada e uma única camada intermediária de  $M$  nós.
- O neurônio  $i$  tem pesos  $w_{1i}, \dots, w_{pi}$  e limiar  $\theta_i$ .
- A saída da rede é uma combinação linear das saídas dos neurônios intermediários com  $\alpha_i$ .



# Teorema da Aproximação Universal (TAU)

- Trata-se de um teorema de **existência**, visto que fornece uma justificativa para a aproximação de funções contínuas.  $\implies$  SUFICIENTE
- Entretanto, ele não afirma que uma única camada é um número **ótimo**.
- Na prática, nem sempre se dispõe de uma função contínua e nem de uma camada intermediária de tamanho qualquer.
- Chester [1990] e Funahashi [1989] defendem o uso de duas camadas intermediárias, tornando a aproximação mais maleável.

# Teorema da Aproximação Universal (TAU)

- Características locais são extraídas na primeira camada.
  - Alguns neurônios na primeira camada são usados para particionar o espaço em várias regiões, e outros aprendem as características locais daquelas regiões.
- Características globais são extraídas na segunda camada.
  - Um neurônio na segunda camada combina as saídas de neurônios da primeira que estão operando numa região particular do espaço de entrada e assim aprende características globais daquela região.

# Sumário

## 1 Introdução

- Modelo de rede MLP

## 2 Treinamento de redes MLP

- O algoritmo Backpropagation
- Processo de aprendizado
- Exemplo

#### 4 Considerações práticas

- Velocidade de aprendizado
- Termo Momentum
- Modos de treinamento
- Critério de parada
- Generalização
- Normalização
- Inicialização

## RESULTS

100

- $O(\log n)$  algorithm: Backpropagation

• **Prevalence** – the proportion of a population that has a disease at a particular point in time

### Method

- Velocidade de aprendizado

- 112

- 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 104

# Velocidade de aprendizado

- O algoritmo BP fornece uma aproximação para a trajetória no espaço dos pesos.
- Quanto menor o valor de  $\eta$ , menores as mudanças nos pesos e mais suave será a trajetória.
  - **Aprendizado lento.**
- Se  $\eta$  é muito grande, o aprendizado torna-se rápido, porém a rede pode tornar-se **instável**.

- Modelo de rede MLP

- O algoritmo Backpropagation
- Processo de aprendizado
- Exemplo

#### 4 Considerações práticas

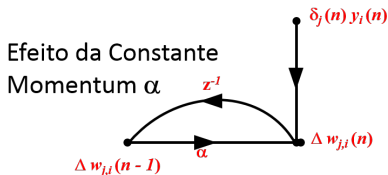
- Velocidade de aprendizado
- **Termo Momentum**
- Modos de treinamento
- Critério de parada
- Generalização
- Normalização
- Inicialização

# Efeito da constante $\alpha$

- É um método simples de aumentar a velocidade do aprendizado e evitar o perigo de instabilidade, como mostrado por Rumelhart *et al.*, 1986.

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) + \alpha \Delta w_{ji}(n-1) \quad (2)$$

- Onde  $\alpha$  é geralmente um número positivo chamado **constante momentum**.



A equação ( $\alpha$ ) é chamada  
**REGRA DELTA**  
**GENERALIZADA**. Se  $\alpha = 0$   
 $\Rightarrow$  REGRA DELTA

## Efeito da constante $\alpha$

- Vamos considerar uma série de tempo com índice  $t$  (de 0 a  $n$ ).
- A equação 2 pode ser vista como uma equação diferencial de primeira ordem em relação a  $\Delta w_{ji}(n)$ . Resolvendo:

$$\Delta w_{ji}(n) = \eta \sum_{t=0}^n \alpha^{n-t} \delta_j(t) y_i(t) \quad (3)$$

- Que representa uma série de tempo comprimido  $n + 1$ . Mas:

$$\begin{aligned} \delta_j(n)y_i(n) &= -\frac{\partial E(n)}{\partial w_{ji}(n)} \\ \therefore \Delta w_{ji}(n) &= -\eta \sum_{t=0}^n \alpha^{n-t} \frac{\partial E(t)}{\partial w_{ji}(t)} \end{aligned} \quad (4)$$



## Efeito da constante $\alpha$

- 1 O ajuste atual  $\Delta w_{ji}(n)$  representa a soma de uma série temporal ponderada exponencialmente convergente  $\implies 0 \leq |\alpha| < 1$
- 2 Quando  $\frac{\partial E(t)}{\partial w_{ji}(t)}$  tem o mesmo sinal algébrico em iterações consecutivas, então a série cresce em magnitude e os pesos são ajustados por uma quantidade grande. Portanto, o BP tende a acelerar a "descida" nas regiões de descida da superfície do erro.
- 3 Quando  $\frac{\partial E(t)}{\partial w_{ji}(t)}$  tem sinais opostos em iterações sucessivas, então a série diminui em magnitude, e  $\Delta w_{ji}(n)$  é atualizado por uma quantidade pequena. Então, a inclusão do termo *momentum* tem o **efeito de estabilização** nas direções em que o sinal oscila.

## Efeito da constante $\alpha$

- Portanto, o termo *momentum* pode ter efeitos benéficos no comportamento do aprendizado do algoritmo. Ele pode evitar que o processo termine em um mínimo local na superfície do erro.
- Observação:** o parâmetro  $\eta$  foi considerado constante.
  - 1  $\eta_{ji}$  **dependente da conexão:** fatos interessantes ocorrem se  $\eta_{ji}$  é tomado diferente em diferentes partes do algoritmo.
  - 2 **Restringir o número de pesos a serem ajustados:**  $\eta_{ji} = 0$  para o peso  $w_{ji}$ .

## Efeito da constante $\alpha$

- Modo segundo o qual as camadas ocultas são interconectadas: no procedimento, supomos que cada camada recebe entradas apenas das unidades da **camada anterior**.
- Não existe uma razão para isso. Se esse não for o caso, existem dois tipos de sinais de erro:
  - Um sinal de erro que resulta de uma comparação direta do sinal de saída daquele neurônio como uma resposta desejada.
  - Um sinal de erro que é passado através de outras unidades cuja ativação ele afeta.'



- Aprendizado BP resulta de muitas apresentações de um conjunto de treinamento de exemplos.
- Uma apresentação **completa** do conjunto de treinamento corresponde a 1 ciclo (*epoch*).
- O processo de aprendizado é repetido ciclo após ciclo, até que os pesos sinápticos e níveis *threshold* se **estabilizem**.
- Tomar os pesos em uma forma aleatória → pesquisa no espaço dos pesos **estocástica**.

# Modo padrão

- **(1) Modo padrão:**

- Atualização nos pesos é feita após a apresentação de cada exemplo de treinamento.
- Um ciclo consistindo de  $N$  exemplos de treinamento, arranjados na ordem:

$$\{[\mathbf{x}_1, \mathbf{d}_1], [\mathbf{x}_2, \mathbf{d}_2], \dots, [\mathbf{x}_N, \mathbf{d}_N]\}$$

- $[\mathbf{x}_1, \mathbf{d}_1] \rightarrow$  cálculos *forward/backward* e atualização dos pesos.
- $[\mathbf{x}_2, \mathbf{d}_2] \rightarrow$  cálculos *forward/backward* e atualização dos pesos.
- $\vdots$
- $[\mathbf{x}_N, \mathbf{d}_N] \rightarrow$  cálculos *forward/backward* e atualização dos pesos.

# Modo padrão

- Dessa forma, a variação média nas mudanças dos pesos é:

$$\begin{aligned}\Delta \hat{w}_{ji} &= \frac{1}{N} \sum_{n=1}^N \Delta w_{ji}(n) \\ &= -\frac{\eta}{N} \sum_{n=1}^N \frac{\partial E(n)}{\partial w_{ji}(n)} \Rightarrow\end{aligned}$$

$$\boxed{\Delta \hat{w}_{ji} = -\frac{\eta}{N} \sum_{n=1}^N e_j(n) \frac{\partial e_j(n)}{\partial w_{ji}(n)}} \quad (5)$$

# Modo *batch*

## • (2) Modo batch:

- Atualização dos pesos é feita depois da apresentação de todos os exemplos de treinamento que constituem um ciclo.
- Para um ciclo particular, função custo com o erro quadrático médio:

$$\mathcal{E}_{av} = \frac{1}{2N} \sum_{n=1}^N \sum_{j \in C} e_j^2(n) \quad (6)$$

- Onde  $C$  denota o conjunto de índices correspondentes aos neurônios da camada de saída e  $e_j$  é o sinal do erro do neurônio  $j$  correspondente ao exemplo de treinamento  $w$ .

$$\Delta w_{ji} = -\eta \frac{\partial \mathcal{E}_{av}}{\partial w_{ji}} \implies$$

$$\Delta w_{ji} = \frac{\eta}{N} \sum_{n=1}^N e_j(n) \frac{\partial e_j(n)}{\partial w_{ji}} \quad (7)$$



# Modos de treinamento - comparação

- Claramente,  $\Delta \hat{w}_{ji}$  é diferente de  $\Delta w_{ji}$ .
  - $\Delta \hat{w}_{ji}$  representa uma **estimativa** de  $\Delta w_{ji}$ .
- Do ponto de vista *online*, o **modo padrão** é preferido. Além disso, os exemplos de treinamento são aleatoriamente apresentados (atualização nos pesos é **estocástica**) → menos provável o algoritmo BP estacionar em um mínimo local.
- Por outro lado, o **modo batch** fornece uma estimativa mais precisa do **vetor gradiente**.
- De qualquer forma, a eficiência dos dois modos depende do problema que se tem em mãos (Hertz, 1991).

# Sumário

- 1 Introdução
  - Modelo de rede MLP
- 2 Treinamento de redes MLP
  - O algoritmo Backpropagation
  - Processo de aprendizado
  - Exemplo
- 3 Teorema da Aproximação Universal
- 4 Considerações práticas
  - Velocidade de aprendizado
  - Termo Momentum
  - Modos de treinamento
  - Critério de parada
  - Generalização
  - Normalização
  - Inicialização

# Critério de parada

- Não se pode, em geral, mostrar a convergência do algoritmo BP, tampouco existem critérios bem definidos para encerrar seu processamento.
- Para formular um critério, devem-se considerar propriedades de mínimo local ou global da superfície de erro.

# Critério de parada

- Seja  $\mathbf{w}^*$  o vetor mínimo local ou global.
- ① Uma condição necessária para  $\mathbf{w}^*$  ser mínimo:
  - O gradiente (derivada de primeira ordem) da superfície de erro em relação a  $\mathbf{w}$  seja zero em  $\mathbf{w} = \mathbf{w}^*$ , isto é,  $\nabla g(w) = 0$  em  $\mathbf{w} = \mathbf{w}^*$ .
  - Diz-se que o algoritmo BP convergiu se a norma do vetor gradiente é menor que um certo  $\epsilon$  pequeno arbitrário.
- ② Função custo  $\mathcal{E}_{av}(w)$  é estacionária em  $\mathbf{w} = \mathbf{w}^*$ .
  - Diz-se que o algoritmo BP convergiu se a taxa de mudança no erro quadrático médio por ciclo é suficientemente pequena.
  - Tipicamente, são consideradas pequenas taxas de mudanças no erro de 0.1% a 1% ou de 0.01%.

# Critério de parada

- Kramer e Sangiovanni-Vicentelli(1989) sugerem um critério de convergência:

O algoritmo BP termina no vetor peso  $w_{final}$  quando  $\|g(w_{final})\| \leq \varepsilon$ , onde  $\varepsilon$  é suficiente pequeno, ou  $\|\varepsilon_{av}(final)\| \leq \tau$  onde  $\tau$  é suficiente pequeno.

# Sumário

- 1 Introdução
  - Modelo de rede MLP
- 2 Treinamento de redes MLP
  - O algoritmo Backpropagation
  - Processo de aprendizado
  - Exemplo
- 3 Teorema da Aproximação Universal
- 4 Considerações práticas
  - Velocidade de aprendizado
  - Termo Momentum
  - Modos de treinamento
  - Critério de parada
  - Generalização
  - Normalização
  - Inicialização

# Generalização

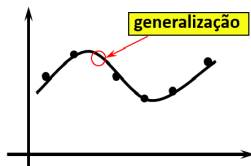
Aprendizado BP

**conjunto treinamento + algoritmo BP  $\Rightarrow$  pesos sinápticos**

GENERALIZAR

"GENERALIZAÇÃO": termo da psicologia.

Processo de aprendizado pode ser visto como um Método de Aproximação de Funções



generalização

: efeito de uma boa aproximação não linear dos dados de entrada, tamanho e eficiência do conjunto treinamento, arquitetura da rede, complexidade física do problema

# Complexidade da rede

- **Problema:** determinar o melhor número de nós na camada intermediária.
- Estatisticamente, esse problema é equivalente a determinar o tamanho do conjunto de parâmetros usado para modelar o conjunto de dados. Existe um limite no tamanho da rede.
- Esse limite deve ser tomado lembrando que é melhor treinar a rede para **produzir a melhor generalização** do que treinar a rede para representar perfeitamente um conjunto de dados.
- Isso pode ser feito usando **validação cruzada**.



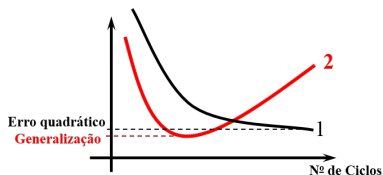
# Validação cruzada

- Conjunto de dados:
  - Treinamento ( 75%)
  - Teste ( 25%)
- Conjunto de treinamento:
  - Um subconjunto para validação do modelo.
  - Um subconjunto para treinamento.
- Validar o modelo em um conjunto diferente do usado para estimá-lo.

# Validação cruzada

- Usa-se o subconjunto de validação para avaliar o desempenho de diferentes candidatos do modelo (diferentes topologias) e, então, escolhe-se uma delas.
- O modelo escolhido é treinado sobre o conjunto de treinamento inteiro e a capacidade de generalização é medida no conjunto de teste.
- A validação cruzada pode ser usada para decidir quando o treinamento de uma rede deve ser encerrado.

# Tamanho do conjunto de treinamento



**Curva 1:** poucos parâmetros (*underfitting*)

**Curva 2:** muitos parâmetros (*overfitting*)

- Em ambos os casos:
  - 1 O desempenho do erro na generalização exibe um mínimo.
  - 2 O mínimo no caso *overfitting* é menor e mais definido.
- Pode-se obter boa generalização se a rede é projetada com muitos neurônios, contanto que o treinamento seja cessado após um número de ciclos correspondente ao mínimo da curva do **erro** obtida na **validação cruzada**.

# Sumário

- 1 Introdução
  - Modelo de rede MLP
- 2 Treinamento de redes MLP
  - O algoritmo Backpropagation
  - Processo de aprendizado
  - Exemplo
- 3 Teorema da Aproximação Universal
- 4 Considerações práticas
  - Velocidade de aprendizado
  - Termo Momentum
  - Modos de treinamento
  - Critério de parada
  - Generalização
  - Normalização
  - Inicialização

# Normalização

## Normalização

Consiste em modificar o processo de aprendizado, de modo a favorecer soluções com determinadas propriedades em detrimento de outras.

- Principais técnicas de normalização:
  - Normalização L1
  - Normalização L2
  - *Early-stopping*

# Normalização

## Normalização L1/L2

Consiste em colocar um termo adicional na função de perda, punindo parametrizações com pesos mais elevados.

# Normalização L1/L2

- Se  $\mathcal{L}(\theta, \mathcal{D})$  é a função perda (e.g. erro quadrático médio),  $\theta$  é o conjunto de parâmetros livres e  $\mathcal{D}$  é um exemplo de treinamento, então a função de perda regularizada será:

$$E(\theta, \mathcal{D}) = \mathcal{L}(\theta, \mathcal{D}) + \lambda R(\theta)$$

- Ou, no nosso caso:

$$E(\theta, \mathcal{D}) = \mathcal{L}(\theta, \mathcal{D}) + \lambda \|\theta\|_p$$

- Onde:

$$\|\theta\|_p = \left( \sum_{j=0}^{|\theta|} |\theta_j|^p \right)^{\frac{1}{p}}$$

- Que é a norma  $L_p$  de  $\theta$ .

# Normalização L1/L2

- $\alpha$  é um hiper-parâmetro (i.e. um parâmetro configurado manualmente) que controla a importância relativa do parâmetro de regularização.
- Valores comuns são  $p = 1$  ou  $p = 2$ , daí o nome **L1/L2**.
- Adicionar um termo de regularização tende a produzir um mapeamento mais suave, uma vez que valores elevados dos parâmetros são penalizados.
- Pelo princípio da Navalha de Occam, será obtida uma função mais simples.
  - Não significa necessariamente que vai generalizar bem.
  - Empiricamente, constata-se que realizar esse tipo de normalização no contexto de redes neurais melhora a generalização, principalmente em bases de dados pequenas.



Consiste em adicionar um coeficiente de paciência e avaliar as parametrizações frequentemente em um conjunto de validação. Quando a melhoria for insignificante ou negativa, retornar a melhor parametrização encontrada até então e encerrar o treinamento.

9. [www.irs.gov](http://www.irs.gov)

- A técnica de *early stopping* requer que particionemos os exemplos em três conjuntos: de treinamento, de validação e de teste.
- O conjunto de treinamento é usado para aplicação de gradiente descendente estocástico na aproximação diferenciável da função perda.
- Enquanto se executa o gradiente descendente, o conjunto de validação é periodicamente consultado para verificar como o modelo está se desempenhando.
- Pode ser adicionado um coeficiente de **paciência**, segundo o qual o algoritmo pára assim que não houver melhora significativa em sucessivas avaliações com o conjunto de validação.

---

● 6. 下列各句，没有语病的一句是（ ）

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210. 211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224. 225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238. 239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252. 253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266. 267. 268. 269. 270. 271. 272. 273. 274. 275. 276. 277. 278. 279. 280. 281. 282. 283. 284. 285. 286. 287. 288. 289. 290. 291. 292. 293. 294. 295. 296. 297. 298. 299. 300. 301. 302. 303. 304. 305. 306. 307. 308. 309. 310. 311. 312. 313. 314. 315. 316. 317. 318. 319. 320. 321. 322. 323. 324. 325. 326. 327. 328. 329. 330. 331. 332. 333. 334. 335. 336. 337. 338. 339. 340. 341. 342. 343. 344. 345. 346. 347. 348. 349. 350. 351. 352. 353. 354. 355. 356. 357. 358. 359. 360. 361. 362. 363. 364. 365. 366. 367. 368. 369. 370. 371. 372. 373. 374. 375. 376. 377. 378. 379. 380. 381. 382. 383. 384. 385. 386. 387. 388. 389. 390. 391. 392. 393. 394. 395. 396. 397. 398. 399. 400. 401. 402. 403. 404. 405. 406. 407. 408. 409. 410. 411. 412. 413. 414. 415. 416. 417. 418. 419. 420. 421. 422. 423. 424. 425. 426. 427. 428. 429. 430. 431. 432. 433. 434. 435. 436. 437. 438. 439. 440. 441. 442. 443. 444. 445. 446. 447. 448. 449. 450. 451. 452. 453. 454. 455. 456. 457. 458. 459. 460. 461. 462. 463. 464. 465. 466. 467. 468. 469. 470. 471. 472. 473. 474. 475. 476. 477. 478. 479. 480. 481. 482. 483. 484. 485. 486. 487. 488. 489. 490. 491. 492. 493. 494. 495. 496. 497. 498. 499. 500. 501. 502. 503. 504. 505. 506. 507. 508. 509. 510. 511. 512. 513. 514. 515. 516. 517. 518. 519. 520. 521. 522. 523. 524. 525. 526. 527. 528. 529. 530. 531. 532. 533. 534. 535. 536. 537. 538. 539. 540. 541. 542. 543. 544. 545. 546. 547. 548. 549. 550. 551. 552. 553. 554. 555. 556. 557. 558. 559. 560. 561. 562. 563. 564. 565. 566. 567. 568. 569. 570. 571. 572. 573. 574. 575. 576. 577. 578. 579. 580. 581. 582. 583. 584. 585. 586. 587. 588. 589. 590. 591. 592. 593. 594. 595. 596. 597. 598. 599. 600. 601. 602. 603. 604. 605. 606. 607. 608. 609. 610. 611. 612. 613. 614. 615. 616. 617. 618. 619. 620. 621. 622. 623. 624. 625. 626. 627. 628. 629. 630. 631. 632. 633. 634. 635. 636. 637. 638. 639. 640. 641. 642. 643. 644. 645. 646. 647. 648. 649. 650. 651. 652. 653. 654. 655. 656. 657. 658. 659. 660. 661. 662. 663. 664. 665. 666. 667. 668. 669. 670. 671. 672. 673. 674. 675. 676. 677. 678. 679. 680. 681. 682. 683. 684. 685. 686. 687. 688. 689. 690. 691. 692. 693. 694. 695. 696. 697. 698. 699. 700. 701. 702. 703. 704. 705. 706. 707. 708. 709. 710. 711. 712. 713. 714. 715. 716. 717. 718. 719. 720. 721. 722. 723. 724. 725. 726. 727. 728. 729. 730. 731. 732. 733. 734. 735. 736. 737. 738. 739. 740. 741. 742. 743. 744. 745. 746. 747. 748. 749. 750. 751. 752. 753. 754. 755. 756. 757. 758. 759. 760. 761. 762. 763. 764. 765. 766. 767. 768. 769. 770. 771. 772. 773. 774. 775. 776. 777. 778. 779. 780. 781. 782. 783. 784. 785. 786. 787. 788. 789. 790. 791. 792. 793. 794. 795. 796. 797. 798. 799. 800. 801. 802. 803. 804. 805. 806. 807. 808. 809. 810. 811. 812. 813. 814. 815. 816. 817. 818. 819. 820. 821. 822. 823. 824. 825. 826. 827. 828. 829. 830. 831. 832. 833. 834. 835. 836. 837. 838. 839. 840. 84

## Inicialização

- O primeiro passo do algoritmo BP é a inicialização da rede.
- Uma boa escolha para os parâmetros livres (pesos sinápticos e *threshold*) podem contribuir significativamente no sucesso do aprendizado.

# Inicialização

- **Informação disponível**
- **Nenhuma informação disponível?**
  - Pesos inicializados aleatoriamente, isto é, inicializar os pesos com valores uniformemente distribuídos em um intervalo pequeno.
- **Escolha errada  $\implies$  saturação prematura**
  - Esse fenômeno se refere a uma situação na qual o erro quadrático permanece constante por um período de tempo, porém continua a diminuir depois que este período é concluído.

# Inicialização

- Fatos interessantes podem ocorrer:
- ❶ Suponha que, para um particular padrão de treinamento, o nível de ativação interna de um neurônio saída tenha um valor cuja magnitude é grande (como a função é *sigmoid*, trata-se de um caso em que  $y = 1$  ou  $y = -1$ ). Em tal caso, diz-se que o neurônio está em **saturação**.
- ❷ Se  $y$  está mais próximo de 1 quando a saída desejada é  $-1$ , ou vice-versa, o neurônio está **incorretamente saturado**.
  - Quando isso ocorre, o ajuste nos pesos será pequeno, embora o erro seja de magnitude grande, e a rede levará um longo tempo para corrigir essa situação (Lee,1991).
- ❸ No estágio inicial do BP, podem existir neurônios não-saturados ou incorretamente saturados.

# Inicialização

- Para os não-saturados  $\rightarrow$  os pesos mudam rapidamente.
- Para os incorretamente saturados  $\rightarrow$  permanecem saturados por algum tempo.
- **Fenômeno da saturação prematura** pode ocorrer, com  $\mathcal{E}$  permanecendo constante.

# Inicialização

- Em Lee(1991), uma fórmula para a probabilidade de **saturação prematura** foi obtida para o **modo batch**.
- A essência dessa fórmula pode ser: [Haykin, 1994]
  - ① **Saturação incorreta** é evitada escolhendo valores iniciais dos pesos sinápticos e níveis *threshold*, uniformemente distribuídos em um intervalo pequeno.
  - ② É menos provável quando o número de neurônios intermediários é mantido baixo.
  - ③ Raramente ocorre quando os neurônios da rede operam em sua regiões lineares.
- Segundo [Haykin,1994], para o **modo padrão** de atualização dos pesos, os resultados mostram uma tendência similar ao **modo batch**.