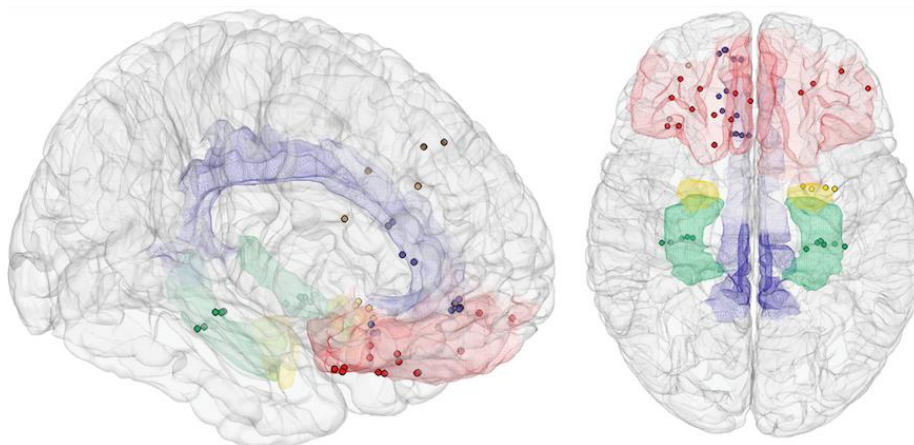


NEUROCIÊNCIA COMPUTACIONAL



Sumário

O que é Neurociência Computacional?.....	3
Como a Mente Computa a Informação?.....	4
A Matemática da Mente.....	6
Aprendizado de Máquina.....	24
A Matemática do Cérebro.....	30
Monitoramento da Atividade Cerebral.....	40
Armazenamento e Recuperação de Memórias.....	43

O que é Neurociência Computacional?

A Neurociência Computacional também conhecida como Neurociência Teórica ou Neurociência Matemática é um subcampo de estudo da Neurociência e da Computação que tem como objetivo estudar o cérebro, o sistema nervoso e a mente através de modelos matemáticos e computacionais. O termo computacional refere-se tanto a parte de desenvolvimento de hardware e/ou software na elaboração desses modelos quanto à parte computacional do próprio cérebro biológico na computação de informações armazenadas e processadas neste.

O termo foi utilizado pela primeira vez em uma conferência de 1985 em Carmel na Califórnia por Eric L. Schwartz, Físico especialista em Sistemas Computacionais Cognitivos. O Termo foi introduzido a pedido da “Systems Development Foundation” que era uma das organizadoras do evento. Até então este campo de estudo era chamado por outros nomes como Modelagem Neural, Teoria do Cérebro ou simplesmente Redes Neurais Artificiais, o que costumava causar confusões entre outras áreas de estudo como a própria Neurociência e a Inteligência Artificial na publicação de artigos científicos. Com a definição do termo o campo de estudo passou a ser relacionado à computação matemática de sistemas cognitivos biológicos através de simulações computacionais.

Como a Mente Computa a Informação?

Nosso cérebro computa a informação por meio do reconhecimento de padrões digitais (através de sinais elétricos) e analógicos (através da anatomia cerebral). Como o cérebro é uma máquina biológica principalmente analógica tendemos a levar muito mais tempo do que computadores (que são 100% digitais) na computação de variáveis matemáticas, mas em contrapartida temos uma capacidade muito superior na interpretação de variáveis externas recebidas como padrões cognitivos por possuímos uma quantidade extremamente grande de sensores corporais na captação desses padrões que são recebidos pelos cinco sentidos (visão, audição, paladar, olfato e tato).

De maneira redundante como nossos sentidos precisam ser “sentidos” por nós, precisamos de um sistema analógico que altere a sua morfologia em tempo de execução para que possamos sentir essas alterações e interpretá-las cognitivamente. Quando essas alterações morfológicas conseguem ser relacionadas a um dos nossos incontáveis conjuntos de variáveis armazenadas ao longo da vida temos o que chamamos de sentimentos, que de maneira computacional nada mais são do que a relação encontrada entre as variáveis que estão sendo recebidas pelos receptores sensoriais e as variáveis armazenadas que estão sendo consultadas pelo cérebro, quando essa relação é encontrada é criada uma nova informação cruzada que nos permite sentir o processamento e armazenamento de dados com a alteração morfológica do cérebro e/ou do corpo.

Do contrário do que muitos pensam, nós não armazenamos e processamos informações somente no cérebro, mas também em diferentes partes do corpo que se comunicam com o cérebro quando essa informação precisa ser consultada.

Essa computação biológica assim como a digital dos computadores poderá conter erros, a principal diferença é que os computadores possuem um sofisticado sistema de correção de erros e nós não. Quando nossas informações são computadas com erro pelo cérebro o resultado é emitido e propagado com erros que podem se expressar através de falsas memórias, esquecimentos, distrações, ilusões e até mesmo apagões e doenças neurológicas em casos mais extremos. Nosso cérebro é uma máquina extremamente complexa, mas complexidade é diferente de perfeição, temos dificuldades e limitações ao estudarmos o cérebro por causa da sua alta complexidade e não por se tratar de uma estrutura perfeita e misteriosa. E complexidade nem sempre é sinônimo de organização, a desordem também pode aumentar a complexidade.

Nosso cérebro também armazena vieses sem qualquer tipo de limitação quando repetidos valores de variáveis são inseridos na mesma região fazendo com que o peso dessa variável aumente perante as outras. Por exemplo: uma pessoa que teve consecutivas experiências ruins com uma determinada etnia terá muito mais chances de desenvolver pensamentos racistas do que uma pessoa que não teve as mesmas experiências. Pessoas que nascem em famílias de uma determinada religião e são constantemente alimentadas com esse tipo de variável muito provavelmente terão as mesmas crenças de seus pais. Outro fator gerador de vieses é a própria sociedade na qual o indivíduo está inserido, sabe-se por exemplo que sociedades com tradição religiosa tendem a ter descendentes religiosos enquanto que em

países de maioria atea na Escandinávia e norte da Europa há uma maior tendência em se gerar descendentes também ateus. Países com tradição em determinados esportes, tipos de música, arte e comportamento também tendem a gerar indivíduos com os mesmos gostos influenciados por esses vieses. Esses vieses da maioria nem sempre refletem a realidade uma vez que se tratam de padrões que foram simplesmente copiados sem qualquer tipo de reflexão ou questionamento. Uma das características encontradas em gênios é a de que eles têm uma grande habilidade em não se deixarem influenciar por vieses da família, sociedade e meio que os cerca, formando pensamentos particulares que muitas vezes são revolucionários.

Uma pessoa que tem medo do escuro ou medo/admiração pelo sobrenatural e pensa nisso constantemente insere cada vez mais variáveis de valores semelhantes na sua mente que formam vieses tão fortes que essa mesma pessoa ao se levantar no meio da noite para tomar um simples copo d'água poderá ver vultos que simplesmente não existem reforçando ainda mais os vieses anteriores. Libertar-se de vieses mentais apesar de tudo pode ser tão fácil quanto assimilá-los bastando consumir conteúdos e ter experiências diferentes das habituais. Uma pessoa por exemplo que seja interessada em política e consome conteúdos tanto de esquerda quanto de direita está muito menos propensa a armazenar vieses irrealistas e ser manipulada por eles do que uma pessoa que vê tudo o que pode de somente um dos lados.

Durante a computação de informação os erros não corrigidos do cérebro também podem nos gerar falsas interpretações que poderão resultar em dúvidas ou até mesmo em certezas de eventos incorretos. Por exemplo, uma pessoa que está acostumada a diferenciar gatos de cachorros não está imune de se confundir quando um deles passa em alta velocidade a sua frente, fazendo com que essa pessoa não saiba qual o tipo de animal que ela viu ou até mesmo podendo concluir que tenha com certeza visto um quando na verdade viu o outro. O mesmo pode ocorrer ao nos depararmos com uma espécie de gato que se parece cachorro ou de cachorro que se parece com gato.

Por tanto quando traduzimos essas computações biológicas em modelos matemáticos como os de **redes neurais artificiais**, por exemplo, as mesmas limitações também são traduzidas. Não raramente muitos desses **modelos de inteligência artificial** interpretam informações de forma errônea e classificam dados de maneira errada, isso por que foram inspirados na biologia que age da mesma forma. Sempre existirão erros tanto em modelos biológicos quanto em modelos matemáticos e temos que estar sempre cientes de que eles estarão presentes quando estamos projetando nossos algoritmos. A diferença primordial é que na computação biológica não conseguimos aplicar essas correções, mas na computação digital sim, ainda que de maneira limitada.

O importante ao se criar modelos computacionais baseados na inteligência biológica não é mitigar o erro para zero, por que isso nem o cérebro humano consegue fazer, mas sim deixar essa taxa de erro abaixo do que seria se a mesma tarefa fosse executada por um ser humano. Por exemplo: não faz nenhum sentido construir um sistema computacional para auxiliar tarefas humanas que acerta menos do que os humanos que executam essa mesma tarefa acertariam. Mas o sistema se torna útil quando mesmo com altas taxas de erro ele consegue acertar mais do que um ser humano acertaria ainda que o erro exista em ambos os casos.

A Matemática da Mente

Para o nosso exemplo iremos pressupor uma situação hipotética de que o nosso cérebro através de estímulos visuais e sonoros recebeu um conjunto de variáveis relativas ao reconhecimento de um gato. Nas variáveis de entrada (inputs) teríamos as características que conseguimos reconhecer no animal naquele momento como pelos, tamanho pequeno em relação ao corpo humano desenvolvido, orelhas curtas, focinho achatado, olhos penetrantes, rabo longo, e o ruído característico de miado. Já na variável de saída (output) que seria o valor expresso como resultado para futuras entradas com características semelhantes teríamos a categoria de nome gato, que seria o nome que ouvimos alguém dizer e que captamos como estímulo sonoro. Vejamos como a nossa mente armazenaria essas variáveis:

Pelos	Pequeno	Orelhas curtas	Focinho achatado	Olhar penetrante	Rabo longo	Miado	Output: GATO
-------	---------	----------------	------------------	------------------	------------	-------	--------------

Agora suponhamos que algum tempo depois tivemos contato com um novo animal que nunca tínhamos visto. Apesar de ter algumas das características do primeiro animal ele causou algum estranhamento na nossa mente, isso por que achamos algumas de suas características muito diferentes das que vimos no animal anterior salvo no nosso cérebro como gato. Esse animal também possui pelos, mas é um pouco maior que o gato, suas orelhas eram bem mais longas, o focinho era longo, o olhar era inocente e não penetrante como o do gato, o rabo era mais curto, porém não tanto e o som emitido era muito diferente, alguém nos disse que esse som correspondia a um latido e que o animal se tratava de um cachorro. Nesse caso nossa mente armazenaria o seguinte grupo de variáveis:

Pelos	Médio	Orelhas longas	Focinho longo	Olhar inocente	Rabo médio	Latido	Output: CACHORRO
-------	-------	----------------	---------------	----------------	------------	--------	------------------

Agora vamos supor que tempos depois um novo animal surgiu no nosso campo de visão e reconhecemos algumas de suas características como: pelos, tamanho grande, orelhas longas, focinho longo, olhar raivoso, rabo médio e som de latido. Neste caso teríamos a seguinte formação mental:

Pelos	Grande	Orelhas longas	Focinho longo	Olhar raivoso	Rabo médio	Latido	Output: ???
-------	--------	----------------	---------------	---------------	------------	--------	-------------

Observe que o conjunto de variáveis não é igual a nenhum dos casos anteriores, então como poderia a nossa mente nos dizer de qual animal se trata? Obviamente você já sabe que se trata de um cachorro por causa do latido, mas seu cérebro só teve contato com dois animais e ainda não sabe que todo gato mia e todo cachorro late. Mesmo que o conjunto de entradas não seja igual a nenhum dos que foram armazenados no cérebro a nossa mente irá comparar as novas variáveis de entrada com os dois conjuntos anteriores e em questão de milionésimos de segundo irá retornar o output do conjunto que mais possui variáveis coincidentes com o nosso

novo conjunto que precisa ser reconhecido. Vamos então simular essa comparação exatamente como à mente humana faria quase que imediatamente.

Valores em memória conhecidos:

Pelos	Pequeno	Orelhas curtas	Focinho achatado	Olhar penetrante	Rabo longo	Miado	Output: GATO
Pelos	Médio	Orelhas longas	Focinho longo	Olhar inocente	Rabo médio	Latido	Output: CACHORRO

Novos valores desconhecidos:

Pelos	Grande	Orelhas longas	Focinho longo	Olhar raivoso	Rabo médio	Latido	Output: ???
-------	--------	----------------	---------------	---------------	------------	--------	-------------

Comparação com o output do tipo gato:

Pelos	Pequeno	Orelhas curtas	Focinho achatado	Olhar penetrante	Rabo longo	Miado	Output: GATO
Pelos	Grande	Orelhas longas	Focinho longo	Olhar raivoso	Rabo médio	Latido	Output: ???

Valores coincidentes: 1.

Comparação com o output do tipo cachorro:

Pelos	Médio	Orelhas longas	Focinho longo	Olhar inocente	Rabo médio	Latido	Output: CACHORRO
Pelos	Grande	Orelhas longas	Focinho longo	Olhar raivoso	Rabo médio	Latido	Output: ???

Valores coincidentes: 5.

Observe que mesmo se tratando de uma raça de cachorro da qual nunca se teve contato ainda assim a sua mente consegue traçar correlações utilizando as variáveis que ela tem armazenadas e reconhecer um padrão. Como o conjunto de variáveis do animal desconhecido possui mais coincidência de valores reconhecidos com cachorro do que com gato o valor de output que a sua mente passará para o seu consciente será **cachorro**.

Daqui para frente esse novo conjunto de dados será salvo na sua mente como um segmento de memória do tipo cachorro com as variáveis pelos, grande, orelhas longas, focinho longo, olhar raivoso, rabo médio e latido. Mas existe uma peculiaridade importante para a mente nesse novo segmento de memória que não existia nos outros: não é a primeira vez que a nossa mente armazena o segmento de memória do tipo cachorro. Então a partir deste momento as variáveis nos seguimentos do tipo cachorro que se repetiram mais de uma vez terão um peso relativo à sua quantidade de repetições, no caso como a classe cachorro está se repetindo pela

segunda vez as suas variáveis coincidentes nos dois segmentos do tipo cachorro terão peso igual a 2. Se fosse a terceira vez que estivéssemos tendo contato com um cachorro elas teriam peso 3 e assim sucessivamente. É por isso que a repetição de tarefas nos faz fixar o conhecimento. Além de aumentar o peso das variáveis, como cada segmento é salvo em regiões diferentes do cérebro, teremos mais regiões disponíveis para a consulta fazendo com que a mente tenha mais opções ao procurar pelo segmento mais próximo, aumentando assim o nível de certeza nas avaliações do nosso consciente. As certezas que temos na vida nada mais são do que valores altos de peso para variáveis que já vimos e armazenamos muitas vezes. Tendemos a não ter certeza das nossas opiniões quando não armazenamos variáveis o suficiente para que elas tivessem pesos relevantes. Vejamos então como ficaria a nossa nova configuração mental, os pesos serão colocados entre parênteses:

Pelos (1)	Pequeno (1)	Orelhas curtas (1)	Focinho achatado (1)	Olhar penetrante (1)	Rabo longo (1)	Miado (1)	Output: GATO
Pelos (2)	Médio (1)	Orelhas longas (2)	Focinho longo (2)	Olhar inocente (1)	Rabo médio (2)	Latido (2)	Output: CACHORRO
Pelos (2)	Grande (1)	Orelhas longas (2)	Focinho longo (2)	Olhar raivoso (1)	Rabo médio (2)	Latido (2)	Output: CACHORRO

Observe que agora os valores que já se repetiram tem um peso superior a aqueles que nunca se repetiram em um segmento de memória. **Não se esqueça de que este é somente um modelo lógico de exemplo, nossa mente trabalha com trilhões de variáveis por segmento que seriam impossíveis de se descrever em um documento, por isso usaremos sempre valores extremamente redutíveis para facilitar a assimilação da lógica matemática.** Dessa vez iremos usar como entrada o seguinte segmento de memória recebido como um novo input:

Pelos	Pequeno	Orelhas curtas	Focinho achatado	Olhar penetrante	Rabo médio	Latido	Output: ???
-------	---------	----------------	------------------	------------------	------------	--------	-------------

Agora para facilitar iremos comparar com todos os seguimentos juntos. Inserindo esse novo segmento de memória aos dados memorizados teremos a seguinte configuração mental:

Pelos (1)	Pequeno (1)	Orelhas curtas (1)	Focinho achatado (1)	Olhar penetrante (1)	Rabo longo (1)	Miado (1)	Output: GATO
Pelos (2)	Médio (1)	Orelhas longas (2)	Focinho longo (2)	Olhar inocente (1)	Rabo médio (2)	Latido (2)	Output: CACHORRO
Pelos (2)	Grande (1)	Orelhas longas (2)	Focinho longo (2)	Olhar raivoso (1)	Rabo médio (2)	Latido (2)	Output: CACHORRO
Pelos	Pequeno	Orelhas curtas	Focinho achatado	Olhar penetrante	Rabo médio	Latido	Output: ???

Valores coincidentes para gato: 5

Valores coincidentes para cachorro: 3.

Neste caso tivemos mais valores coincidentes na classe de rótulo gato (5) do que na classe de rótulo cachorro (somente 3). Mas lembre-se de que nas classes com o rótulo cachorro alguns dos blocos de memória possuem peso 2. No caso temos 5 blocos de memória categorizados como cachorro que possuem peso 2 e desses 5, 3 deles foram reconhecidos como valores coincidentes, quando ocorre empate a mente considera apenas o segmento mais recente, logo faremos: 3 (valores coincidentes) + 3 (valores com o dobro do peso) = 6. Como o peso é de 2 os

valores coincidentes serão calculados em dobro. Ou se você preferir basta somar os pesos dos valores coincidentes. Neste caso o resultado final seria:

Valores coincidentes para gato: 5

Valores coincidentes para cachorro: 6.

Logo o valor de output emitido pela mente será o de que o animal foi reconhecido como sendo um cachorro, pois o valor de coincidência da classe cachorro é superior ao da classe gato. Perceba que não há escolha, não há decisão e a matemática deixa isso muito claro, a sua mente decide tudo para você mesmo você achando erroneamente que decidiu definir a nova entrada como cachorro. Não há qualquer tipo de controle individual na seleção dessas variáveis, a seleção é totalmente realizada pelo arranjo em que elas se encontram no cérebro. Agora o nosso novo seguimento de memória será salvo com a seguinte configuração:

Pelos	Pequeno	Orelhas curtas	Focinho achatado	Olhar penetrante	Rabo médio	Latido	Output: CACHORRO
-------	---------	----------------	------------------	------------------	------------	--------	------------------

E os valores que novamente foram coincidentes na classe cachorro serão acrescidos em 1 reforçando ainda mais as características de cachorros na nossa memória.

Pelos (1)	Pequeno (1)	Orelhas curtas (1)	Focinho achatado (1)	Olhar penetrante (1)	Rabo longo (1)	Miado (1)	Output: GATO
Pelos (3)	Médio (1)	Orelhas longas (2)	Focinho longo (2)	Olhar inocente (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Pelos (3)	Grande (1)	Orelhas longas (2)	Focinho longo (2)	Olhar raivoso (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Pelos (3)	Pequeno (1)	Orelhas curtas (1)	Focinho achatado (1)	Olhar penetrante (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO

Para o exemplo a seguir iremos acrescentar um novo segmento de memória do tipo CACHORRO com algumas variáveis desconhecidas representadas pelo valor “Nulo”:

Pelos (1)	Pequeno (1)	Orelhas curtas (1)	Focinho achatado (1)	Olhar penetrante (1)	Rabo longo (1)	Miado (1)	Output: GATO
Pelos (3)	Médio (1)	Orelhas longas (2)	Focinho longo (2)	Olhar inocente (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Pelos (3)	Grande (1)	Orelhas longas (2)	Focinho longo (2)	Olhar raivoso (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Nulo (1)	Nulo (1)	Orelhas curtas (2)	Focinho achatado (2)	Nulo (1)	Nulo (1)	Nulo (1)	Output: CACHORRO
Pelos (3)	Pequeno (1)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO

Vamos agora simular o recebimento por nossas vias sensoriais do seguinte segmento de memória:

Pelos	Pequeno	Orelhas curtas	Focinho achatado	Olhar penetrante	Rabo longo	Miado	Output: ???
-------	---------	----------------	------------------	------------------	------------	-------	-------------

Realizando a comparação com o nosso conjunto de segmentos de memória teremos:

Pelos (1)	Pequeno (1)	Orelhas curtas (1)	Focinho achatado (1)	Olhar penetrante (1)	Rabo longo (1)	Miado (1)	Output: GATO
Pelos (3)	Médio (1)	Orelhas longas (2)	Focinho longo (2)	Olhar inocente (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Pelos (3)	Grande (1)	Orelhas longas (2)	Focinho longo (2)	Olhar raivoso (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Nulo (1)	Nulo (1)	Orelhas curtas (2)	Focinho achatado (2)	Nulo (1)	Nulo (1)	Nulo (1)	Output: CACHORRO
Pelos (3)	Pequeno (1)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Pelos	Pequeno	Orelhas curtas	Focinho achatado	Olhar penetrante	Rabo longo	Miado	Output: ???

Na ocasião atual você deve estar se perguntando... como será possível sair um gato daí? Somando os pesos do melhor resultado para gato teremos 7 que corresponde ao primeiro segmento onde acertamos todos os valores. Em contrapartida no melhor resultado para cachorro mesmo com dois valores errados se somarmos os pesos dos acertos teremos 9, logo a saída deveria ser cachorro, correto? Errado! Neste caso comparativo quando a mente encontra um segmento de memória em que todas as variáveis são iguais a todas as variáveis do segmento de entrada ela simplesmente ignora os demais segmentos e retorna como output a saída do segmento com 100% de correspondência, interrompendo o processamento de informação de qualquer segmento que venha depois deste ponto. Como o segmento com 100% de correspondência é o que possui o output com valor gato, então será gato o output de resultado do nosso novo segmento. A mente sempre inicia a computação em ordem decrescente/descendente, isso significa que ela sempre começa a comparar do segmento mais recente para o mais antigo, dando prioridade para as memórias novas em detrimento das antigas. É por isso que para nós é mais fácil lembrar de coisas que acabaram de acontecer do que de coisas que já aconteceram a dias ou até mesmo anos. Se nesse processo comparativo for encontrado um segmento de correspondência total, ou seja, de 100%, nada que seja mais antigo a partir deste segmento importará, fazendo com que a computação seja interrompida neste ponto emitindo imediatamente a resposta do segmento corrente. Mas e se existirem dois ou mais segmentos com correspondência total e saídas diferentes? Por exemplo, imaginemos a situação a seguir:

1	Pelos (4)	Pequeno (2)	Orelhas curtas (3)	Focinho achatado (3)	Olhar penetrante (2)	Rabo longo (1)	Som desconhecido (1)	Output: CACHORRO
2	Pelos (1)	Pequeno (1)	Orelhas curtas (1)	Focinho achatado (1)	Olhar penetrante (1)	Rabo longo (1)	Som desconhecido (1)	Output: GATO
3	Pelos (4)	Médio (1)	Orelhas longas (2)	Focinho longo (2)	Olhar inocente (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
4	Nulo (1)	Nulo (1)	Orelhas curtas (2)	Focinho achatado (3)	Nulo (1)	Nulo (1)	Nulo (1)	Output: CACHORRO
5	Pelos (4)	Grande (1)	Orelhas longas (2)	Focinho longo (2)	Olhar raivoso (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
6	Pelos (4)	Pequeno (2)	Orelhas curtas (3)	Focinho achatado (3)	Olhar penetrante (2)	Rabo médio (3)	Latido (3)	Output: CACHORRO
	Pelos	Pequeno	Orelhas curtas	Focinho achatado	Olhar penetrante	Rabo longo	Som desconhecido	Output: ???

Para facilitar o entendimento inserimos uma coluna com IDs de 1 a 6 para cada segmento memorizado, os identificadores de menor valor correspondem aos segmentos que foram inseridos primeiro na memória, ou seja, correspondem às memórias mais antigas, e os identificadores de maior valor correspondem aos segmentos que foram inseridos por último na memória, ou seja, correspondem às memórias mais recentes. Como temos mais de um segmento de correspondência total (no caso 2, mas poderiam ser múltiplos) a resposta ainda assim seria gato por que o ID do gato é igual a 2 e por tanto maior do que o do cachorro que é 1. Logo como o segmento do gato é mais recente será ele o utilizado. No momento em que a mente encontra o primeiro segmento de correspondência total que aqui estava na posição 2, ela para a computação de informação e tudo o que for menor que 2 será ignorado. Se ao invés de 2 o primeiro ID de correspondência total estivesse por exemplo na posição 150, tudo o que fosse menor que 150 seria simplesmente desconsiderado.

Essas memórias antigas só poderiam ser acessadas se o indivíduo tivesse essa intenção e fizesse muita força para acessá-las e a capacidade para acessá-las dependeria do poder de processamento do seu cérebro, ou seja da quantidade de neurônios que ele tem a sua disposição. A quantidade de neurônios no cérebro seria o equivalente a quantidade de transistores em um processador. É por esse motivo que algumas atividades como o consumo de certos tipos de drogas entorpecentes que matam/deletam neurônios tornam os usuários lentos no raciocínio e com a memória cada vez mais fraca, é como se você estivesse retirando transistores do seu processador. Queime alguns transistores do processador do seu computador e verá como ele ficará bem mais lento e menos eficiente. De forma automática sem que o indivíduo precisasse ter a intenção de recuperar tais valores, ele só poderia acessar até o nível 150 (ou nível 2 se considerarmos o conjunto da tabela anterior). Tudo o que estiver abaixo desse nível fará parte de um conjunto de segmentos que na neurociência chamamos de inconsciente. E tudo o que estiver acima desse nível fará parte do conjunto de segmentos do consciente. É assim que é feita a separação lógica entre consciente e inconsciente no cérebro.

Já parou para pensar por que atividades como caminhar, respirar, ou até mesmo as funções fisiológicas mais baixas como o bater do coração, o funcionamento do fígado, rins, intestino e etc. são executadas de forma inconsciente? Isso ocorre por que essas funções foram armazenadas nos segmentos de memória mais antigos do nosso corpo, tais variáveis foram criadas quando ainda estávamos no útero de nossas mães e seus IDs de memória possuem valores extremamente baixos. Valores muito baixos são impossíveis de se acessar, é por isso que não conseguimos controlar nossos órgãos de maneira consciente.

Um fator importante a se destacar é que o conjunto de variáveis do inconsciente quando não for extremamente antigo como o das funções vitais por exemplo, só poderá ser acessado de forma intencional, ou seja, fazendo força com o objetivo de lembrar. Já as variáveis do consciente, que são as mais recentes podem ser naturalmente acessadas de maneira não intencional e totalmente automática.

Vamos agora voltar para a nossa tabela original e considerar que temos o seguinte conjunto de variáveis memorizadas após a inserção do último segmento do tipo gato:

Pelos (2)	Pequeno (2)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (2)	Rabo longo (2)	Miado (2)	Output: GATO
Pelos (3)	Médio (1)	Orelhas longas (2)	Focinho longo (2)	Olhar inocente (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Pelos (3)	Grande (1)	Orelhas longas (2)	Focinho longo (2)	Olhar raivoso (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Nulo (1)	Nulo (1)	Orelhas curtas (2)	Focinho achatado (2)	Nulo (1)	Nulo (1)	Nulo (1)	Output: CACHORRO
Pelos (3)	Pequeno (1)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Pelos (2)	Pequeno (2)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (2)	Rabo longo (2)	Miado (2)	Output: GATO

Observe que agora que temos mais de um exemplo para cada classe é possível perceber que todo gato mia e todo cachorro late. O valor “Nulo” será ignorado por que é interpretado como uma informação não recebida, diferente do que seria se fosse um “Som desconhecido” onde escutaríamos algo, mas sem conseguir distinguir entre um som e outro. Então nossa mente irá adicionar um marcador de separação nas variáveis que correspondem a miado e a latido desse nosso conjunto de exemplo. Dessa forma toda vez que for realizado o processo de comparação nesse conjunto ele será iniciado a partir das variáveis separadoras e se por um acaso o indivíduo dono dessa mente ouvir um miado sem sequer ver o animal ele saberá que se trata

de um gato, da mesma forma ele saberá que se trata de um cachorro toda vez que ouvir um latido, eliminando assim a necessidade de processamento das outras variáveis que só serão utilizadas quando os sentidos que as envolvem como tato e visão por exemplo forem acionados.

Pelos (2)	Pequeno (2)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (2)	Rabo longo (2)	Miado (2)	Output: GATO
Pelos (3)	Médio (1)	Orelhas longas (2)	Focinho longo (2)	Olhar inocente (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Pelos (3)	Grande (1)	Orelhas longas (2)	Focinho longo (2)	Olhar raivoso (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Nulo (1)	Nulo (1)	Orelhas curtas (2)	Focinho achatado (2)	Nulo (1)	Nulo (1)	Nulo (1)	Output: CACHORRO
Pelos (3)	Pequeno (1)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Pelos (2)	Pequeno (2)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (2)	Rabo longo (2)	Miado (2)	Output: GATO

Nossos marcadores de separação serão aqui representados com uma coloração mais escura para facilitar o entendimento. A partir desse ponto as comparações seriam feitas em ordem decrescente do mais atual para o mais antigo e iniciando na variável sonora de miado e latido. Ou seja, na nossa tabela a comparação seguiria a ordem de baixo para cima e da direita para a esquerda. Se o valor da primeira coluna da direita for um valor conhecido e corresponder a miado o resultado será sempre gato, se corresponder a latido o resultado será sempre cachorro e o restante das variáveis só será consultado se não for possível à identificação do miado ou do latido.

Confira como ficaria a entrada de um novo segmento de memória com a seguinte configuração:

Sem pelos	Médio	Orelhas curtas	Focinho achatado	Olhar penetrante	Rabo longo	Miado	Output: ???
-----------	-------	----------------	------------------	------------------	------------	-------	-------------

Neste caso como o miado é um valor de separação todas as outras variáveis no segmento serão ignoradas e o output será imediatamente definido como gato. Como as outras variáveis não foram avaliadas na computação, elas permanecerão com o peso inalterado, mesmo aquelas que seriam coincidentes caso a computação completa fosse processada.

Pelos (2)	Pequeno (2)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (2)	Rabo longo (2)	Miado (2)	Output: GATO
Pelos (3)	Médio (1)	Orelhas longas (2)	Focinho longo (2)	Olhar inocente (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Pelos (3)	Grande (1)	Orelhas longas (2)	Focinho longo (2)	Olhar raivoso (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Nulo (1)	Nulo (1)	Orelhas curtas (2)	Focinho achatado (2)	Nulo (1)	Nulo (1)	Nulo (1)	Output: CACHORRO
Pelos (3)	Pequeno (1)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Pelos (2)	Pequeno (2)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (2)	Rabo longo (2)	Miado (2)	Output: GATO
Sem pelos (1)	Médio (1)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (2)	Rabo longo (2)	Miado (2)	Output: GATO

Porém se o seguimento a seguir fosse recebido à computação seria processada de forma completa varrendo todas as variáveis por segmento (de baixo para cima, da direita para esquerda). Neste caso o processamento seria completo por que a variável de separação não pode ser reconhecida...

Sem pelos	Médio	Orelhas curtas	Focinho achatado	Olhar penetrante	Rabo longo	Som desconhecido	Output: ???
-----------	-------	----------------	------------------	------------------	------------	------------------	-------------

Para facilitar a visualização do cálculo iremos utilizar uma coluna na extremidade esquerda com identificadores e uma coluna na extremidade direita com a contagem dos pesos. Vejamos como ficaria o processamento:

1	Pelos (2)	Pequeno (2)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (2)	Rabo longo (2)	Miado (2)	Output: GATO	08
2	Pelos (3)	Médio (1)	Orelhas longas (2)	Focinho longo (2)	Olhar inocente (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO	01
3	Pelos (3)	Grande (1)	Orelhas longas (2)	Focinho longo (2)	Olhar raivoso (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO	00
4	Nulo (1)	Nulo (1)	Orelhas curtas (2)	Focinho achatado (2)	Nulo (1)	Nulo (1)	Nulo (1)	Output: CACHORRO	04
5	Pelos (3)	Pequeno (1)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO	05
6	Pelos (2)	Pequeno (2)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (2)	Rabo longo (2)	Miado (2)	Output: GATO	08
7	Sem pelos (1)	Médio (1)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (2)	Rabo longo (2)	Miado (2)	Output: GATO	10
	Sem pelos	Médio	Orelhas curtas	Focinho achatado	Olhar penetrante	Rabo longo	Som desconhecido	Output: ???	

Observe que o segmento com a maior soma de pesos é o segmento na posição 7 com um total de 10 na contagem dos pesos. Como este segmento pertence à classe gato então nosso valor de resposta será gato.

Pelos (2)	Pequeno (2)	Orelhas curtas (3)	Focinho achatado (3)	Olhar penetrante (3)	Rabo longo (3)	Miado (2)	Output: GATO
Pelos (3)	Médio (1)	Orelhas longas (2)	Focinho longo (2)	Olhar inocente (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Pelos (3)	Grande (1)	Orelhas longas (2)	Focinho longo (2)	Olhar raivoso (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Nulo (1)	Nulo (1)	Orelhas curtas (2)	Focinho achatado (2)	Nulo (1)	Nulo (1)	Nulo (1)	Output: CACHORRO
Pelos (3)	Pequeno (1)	Orelhas curtas (2)	Focinho achatado (2)	Olhar penetrante (1)	Rabo médio (3)	Latido (3)	Output: CACHORRO
Pelos (2)	Pequeno (2)	Orelhas curtas (3)	Focinho achatado (3)	Olhar penetrante (3)	Rabo longo (3)	Miado (2)	Output: GATO
Sem pelos (2)	Médio (2)	Orelhas curtas (3)	Focinho achatado (3)	Olhar penetrante (3)	Rabo longo (3)	Miado (2)	Output: GATO
Sem pelos (2)	Médio (2)	Orelhas curtas (3)	Focinho achatado (3)	Olhar penetrante (3)	Rabo longo (3)	Som desconhecido (1)	Output: GATO

Neste momento já é possível constatar que conseguimos identificar gatos mesmo estes estando sem pelos e com um som desconhecido, simplesmente interpretando as variáveis sensoriais de tato e/ou visão, obviamente se tivéssemos o estímulo sonoro de miado usaríamos também a audição para a captura, processamento e interpretação dos dados.

Note também que agora não temos mais as nossas variáveis de separação que perderam esse status no momento em que um segmento de som desconhecido foi inserido na memória. Isso acontece por que teoricamente um som desconhecido poderia ser tanto um miado quanto um latido o que alteraria a separação das duas classes existentes (gato e cachorro) em dois sons diferentes (miado e latido). Se estivéssemos trabalhando com três tipos de classes além de gato e cachorro por exemplo, a variável só entraria em status de separação se existissem somente três valores possíveis além do nulo, um para cada classe em uma das posições de memória. Assumindo que existisse mais uma classe do tipo lobo no nosso conjunto de dados e que a separação ocorresse na posição de memória referente ao som então o status de variável de separação para o som só existiria enquanto todo gato miasse, todo cachorro latisse e todo lobo uivasse. Logo o número de valores possíveis na variável de separação deve ser sempre igual ao número possível de classes.

Tem um aspecto fundamental da neurobiologia envolvendo os processos de armazenamento e processamento de memória que até agora ignoramos para simplificar a aprendizagem que é o fato de que o cérebro não utiliza valores categóricos como fizemos até agora, mas ao invés disso ele utiliza cargas elétricas que são geradas através de reações bioquímicas. Então para representar essas cargas elétricas usaremos números de menor valor que representarão as

cargas de menor intensidade e números de maior valor para cargas de maior intensidade, obviamente também existirão números intermediários para cargas mais equilibradas. O valor dessas cargas elétricas não é binário como acontece nos computadores. Apesar do cérebro também computar informação de forma digital binária, a maior parte da computação é não binária. Sabemos por exemplo que a cor vermelha utiliza cargas elétricas de maior intensidade por que evoluímos para que o vermelho nos chamasse mais a atenção do que as outras cores por ser a mesma cor do sangue que aparece nos ferimentos e por isso poderia representar um alerta de perigo. Toques mais intensos, cheiros mais fortes, barulhos mais altos e etc. tudo isso irá influenciar na intensidade das cargas que serão processadas. Por isso a maioria dos algoritmos de inteligência artificial usam números e não letras por exemplo.

Vamos então usar os seguintes valores de referência para o nosso exemplo:

Interpretação	Carga elétrica
Nulo	00
Sem pelos	01
Pelos	02
Pequeno	03
Médio	04
Grande	05
Orelhas curtas	06
Orelhas longas	07
Focinho achatado	08
Focinho longo	09
Olhar inocente	10
Olhar penetrante	11
Olhar raivoso	12
Rabo médio	13
Rabo longo	14
Som desconhecido	15
Miado	16
Latido	17
GATO	18
CACHORRO	19

O cérebro altera a intensidade das cargas elétricas de acordo com o nível de sensibilidade que é sentido pelos terminais sensoriais, mas aqui para o nosso exemplo iremos apenas escolher valores arbitrários para facilitar o entendimento.

Teremos a seguinte configuração final:

2 (2)	3 (2)	6 (3)	8 (3)	11 (3)	14 (3)	16 (2)	Output: 18
2 (3)	4 (1)	7 (2)	9 (2)	10 (1)	13 (3)	17 (3)	Output: 19
2 (3)	5 (1)	7 (2)	9 (2)	12 (1)	13 (3)	17 (3)	Output: 19
0 (1)	0 (1)	6 (2)	8 (2)	0 (1)	0 (1)	0 (1)	Output: 19
2 (3)	3 (1)	6 (2)	8 (2)	11 (1)	13 (3)	17 (3)	Output: 19
2 (2)	3 (2)	6 (3)	8 (3)	11 (3)	14 (3)	16 (2)	Output: 18
1 (2)	4 (2)	6 (3)	8 (3)	11 (3)	14 (3)	16 (2)	Output: 18
1 (2)	4 (2)	6 (3)	8 (3)	11 (3)	14 (3)	15 (1)	Output: 18

Da mesma forma que estávamos computando informações categóricas no formato texto, poderíamos facilmente também computar as mesmas informações no formato numérico utilizando valores referentes a intensidades de cargas elétricas exatamente como acontece nos nossos processos mentais.

Para simplificar iremos remover a quarta linha que contém valores nulos, supondo que a última linha não existisse e que a coluna das variáveis de som estivesse com o status de variável separativa nós poderíamos representar as variáveis de separação com o expoente 2 e as variáveis de status convencional com o expoente 1 deixando a nossa tabela com o seguinte aspecto:

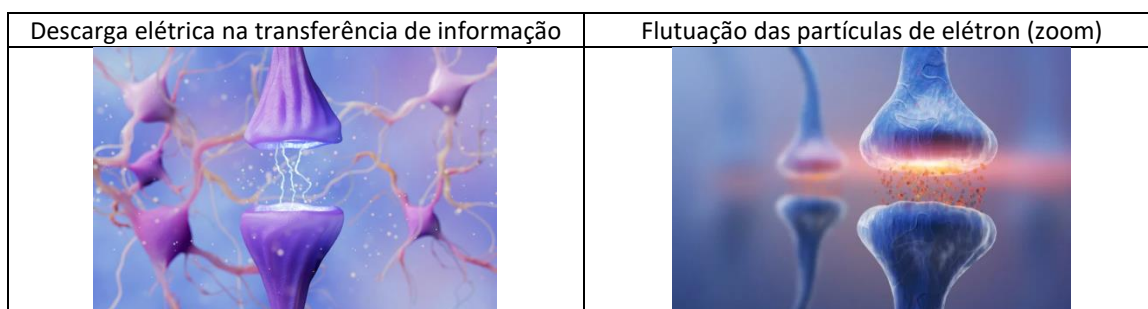
2 (2) ¹	3 (2) ¹	6 (3) ¹	8 (3) ¹	11 (3) ¹	14 (3) ¹	16 (2) ²	Output: 18
2 (3) ¹	4 (1) ¹	7 (2) ¹	9 (2) ¹	10 (1) ¹	13 (3) ¹	17 (3) ²	Output: 19
2 (3) ¹	5 (1) ¹	7 (2) ¹	9 (2) ¹	12 (1) ¹	13 (3) ¹	17 (3) ²	Output: 19
2 (3) ¹	3 (1) ¹	6 (2) ¹	8 (2) ¹	11 (1) ¹	13 (3) ¹	17 (3) ²	Output: 19
2 (2) ¹	3 (2) ¹	6 (3) ¹	8 (3) ¹	11 (3) ¹	14 (3) ¹	16 (2) ²	Output: 18
1 (2) ¹	4 (2) ¹	6 (3) ¹	8 (3) ¹	11 (3) ¹	14 (3) ¹	16 (2) ²	Output: 18

Dessa forma nós poderíamos facilmente reconhecer quais seriam as variáveis de separação para assim identificarmos por onde o cálculo deverá ser iniciado. Neste caso de baixo para cima iniciando na coluna onde os expoentes são iguais a 2. Agora temos uma tabela que usa tanto valores digitais numéricos quanto valores digitais binários (1 e 2 nos expoentes).

Acrescentando a coluna com os índices de identificação de segmento a nossa tabela terá o seguinte aspecto visual:

1	2 (2) ¹	3 (2) ¹	6 (3) ¹	8 (3) ¹	11 (3) ¹	14 (3) ¹	16 (2) ²	Output: 18
2	2 (3) ¹	4 (1) ¹	7 (2) ¹	9 (2) ¹	10 (1) ¹	13 (3) ¹	17 (3) ²	Output: 19
3	2 (3) ¹	5 (1) ¹	7 (2) ¹	9 (2) ¹	12 (1) ¹	13 (3) ¹	17 (3) ²	Output: 19
4	2 (3) ¹	3 (1) ¹	6 (2) ¹	8 (2) ¹	11 (1) ¹	13 (3) ¹	17 (3) ²	Output: 19
5	2 (2) ¹	3 (2) ¹	6 (3) ¹	8 (3) ¹	11 (3) ¹	14 (3) ¹	16 (2) ²	Output: 18
6	1 (2) ¹	4 (2) ¹	6 (3) ¹	8 (3) ¹	11 (3) ¹	14 (3) ¹	16 (2) ²	Output: 18

Nosso cérebro não realiza correções de erro durante o processo computacional da informação e esses erros ocorrem principalmente devido as características quânticas do elétron. De acordo com a teoria da informação quântica não devemos tratar a natureza do elétron de forma determinística, mas sim de forma probabilística. Essa característica não exata do elétron poderá gerar alguns valores estocásticos (aleatórios) nas flutuações quânticas ocorridas na sinapse cerebral. A sinapse ocorre na região entre os terminais do axônio de um neurônio e os dendritos de outro. Por essa região é transmitida a informação em forma de sinal elétrico como pode ser observado na figura abaixo:



Os neurônios não se tocam completamente então os elétrons da corrente elétrica passam por um estágio de flutuação quântica entre os neurônios que geram algumas imprecisões matemáticas. Você já se perguntou: por que estou pensando isso? Ou... de onde veio essa ideia? Por que eu fiz isso? E etc. Isso se deve a essa imprecisão estocástica na computação da informação que quando relacionada às diversas outras variáveis da mente podem resultar tanto em ideias, pensamentos e ações benéficas quanto maléficas para o indivíduo dependendo do ponto de vista. Os defensores do livre arbítrio acreditam que essa flutuação quântica poderia não ser estocástica, mas sim determinada por nós mesmos a nível quântico em uma espécie de decisão que antecederia a própria construção do pensamento. **Mas que fique claro que isso não passa de uma suposição hipotética sem base científica.** Até onde sabemos com o que a Física Quântica/Mecânica Quântica pode nos explicar é que os resultados de flutuações quânticas serão sempre estocásticos e nunca determinísticos, logo o consenso da academia é de que livre arbítrio de fato não existe.

Para simular essa imprecisão quântica nós iremos inserir uma notação decimal aleatória aos nossos números. Em uma estrutura mental real essa precisão tende ao infinito, mas aqui iremos utilizar apenas duas casas decimais para efeitos didáticos. Então teremos algo parecido com o que consta na tabela abaixo:

1	2.45 (2) ¹	3.02 (2) ¹	6.75 (3) ¹	8.46 (3) ¹	11.44 (3) ¹	14.20 (3) ¹	16.07 (2) ¹	18
2	2.32 (3) ¹	4.37 (1) ¹	7.99 (2) ¹	9.67 (2) ¹	10.98 (1) ¹	13.45 (3) ¹	17.53 (3) ¹	19
3	2.23 (3) ¹	5.72 (1) ¹	7.81 (2) ¹	9.03 (2) ¹	12.50 (1) ¹	13.57 (3) ¹	17.66 (3) ¹	19
4	2.17 (3) ¹	3.33 (1) ¹	6.11 (2) ¹	8.18 (2) ¹	11.77 (1) ¹	13.27 (3) ¹	17.40 (3) ¹	19
5	2.01 (2) ¹	3.15 (2) ¹	6.51 (3) ¹	8.12 (3) ¹	11.13 (3) ¹	14.80 (3) ¹	16.08 (2) ¹	18
6	1.55 (2) ¹	4.28 (2) ¹	6.79 (3) ¹	8.55 (3) ¹	11.19 (3) ¹	14.04 (3) ¹	16.02 (2) ¹	18

Como já sabemos que a última coluna representa os valores de saída então iremos ocultar a palavra “output” para facilitar a visualização. Também alteramos o status da última coluna de 2 para 1 pois agora todos os valores para essa posição de memória são diferentes.

Agora que temos valores numéricos não precisamos mais tratar a informação como dados categóricos, ao invés disso iremos calcular a diferença exata entre as cargas, exatamente como é feito no cérebro.

Vamos utilizar por exemplo o seguinte segmento de entrada:

2.41	3.50	6.99	8.99	11.02	14.21	16.15	???
------	------	------	------	-------	-------	-------	-----

Agora nós iremos subtrair cada segmento memorizado do nosso segmento de entrada, para deixar o resultado positivo e calcular a proximidade entre eles nós iremos elevar as diferenças ao quadrado (para positivar) e depois extrair a raiz quadrada da exponenciação (para calcular a distância). Para facilitar o entendimento iremos ocultar os pesos para visualiza-los somente no momento em que eles forem utilizados. O status também será omitido uma vez que todas as variáveis possuem o mesmo status de valor 1. Vamos fazer o nosso cálculo por etapas para que você possa entender melhor.

Etapas 1, subtração (calculando a diferença):

1	0.04	-0.48	-0.24	-0.53	0.42	-0.01	-0.08	18
2	-0.09	0.87	1.00	0.68	-0.04	-0.76	1.38	19
3	-0.18	2.22	0.82	0.04	1.48	-0.64	1.51	19
4	-0.24	-0.17	-0.88	-0.81	0.75	-0.94	1.25	19
5	-0.40	-0.35	-0.48	-0.87	0.11	0.59	-0.07	18
6	-0.86	0.78	-0.20	-0.44	0.17	-0.17	-0.13	18

Etapas 2, exponenciação (deixando os valores positivos):

1	0.00	0.23	0.06	0.28	0.18	0.00	0.01	18
2	0.01	0.76	1.00	0.46	0.00	0.58	1.90	19
3	0.03	4.93	0.67	0.00	2.19	0.41	2.28	19
4	0.06	0.03	0.77	0.66	0.56	0.88	1.56	19
5	0.16	0.12	0.23	0.76	0.01	0.35	0.00	18
6	0.74	0.61	0.04	0.19	0.03	0.03	0.02	18

Agora que todos os números estão positivos, como eles foram todos elevados ao quadrado, iremos extrair a raiz quadrada para retorná-los a proporção original.

Etapa 3, radiciação (voltando os valores para a proporção original):

1	0.04	0.48	0.24	0.53	0.42	0.01	0.08	18
2	0.09	0.87	1.00	0.68	0.04	0.76	1.38	19
3	0.18	2.22	0.82	0.04	1.48	0.64	1.51	19
4	0.24	0.17	0.88	0.81	0.75	0.94	1.25	19
5	0.40	0.35	0.48	0.87	0.11	0.59	0.07	18
6	0.86	0.78	0.20	0.44	0.17	0.17	0.13	18

Agora já temos a diferença exata entre os nossos valores em memória e os novos valores de entrada. Até aqui se você já teve algum contato com algoritmos de aprendizado de máquina deve ter percebido que essa estrutura de cálculo se assemelha muito a alguns algoritmos de **machine learning** como por exemplo o k-nearest neighbors (knn) ou as árvores de decisão se compararmos com o início da nossa explicação quando tínhamos apenas dados categóricos textuais, isso não é por acaso pois a maioria desses algoritmos utilizados em aprendizado de máquina foram elaborados com base em abstrações matemáticas do funcionamento da mente. Mas não se engane, como veremos a seguir este é um cálculo bem mais completo e diferente, os algoritmos de aprendizado de máquina são elaborados com apenas alguns dos aspectos fundamentais da lógica mental por motivos de performance já que se o cálculo fosse aplicado exatamente como é feito na mente humana o tempo de processamento inviabilizaria o seu uso.

A seguir iremos utilizar os pesos, mas dessa vez como não estamos aplicando um cálculo de comparação categórica como antes, nós iremos dividir as diferenças pelos pesos, ou seja, quanto maior for o peso menor será a diferença. A seguir temos a nossa tabela de pesos:

1	2	2	3	3	3	3	2	18
2	3	1	2	2	1	3	3	19
3	3	1	2	2	1	3	3	19
4	3	1	2	2	1	3	3	19
5	2	2	3	3	3	3	2	18
6	2	2	3	3	3	3	2	18

Etapa 4, divisão, dividindo a tabela de radiciação na etapa 3 pela tabela de pesos teremos:

1	0.02	0.24	0.08	0.176666667	0.14	0.003333333	0.04	18
2	0.03	0.87	0.5	0.34	0.04	0.253333333	0.46	19
3	0.06	2.22	0.41	0.02	1.48	0.213333333	0.503333333	19
4	0.08	0.17	0.44	0.405	0.75	0.313333333	0.416666667	19
5	0.2	0.175	0.16	0.29	0.036666667	0.196666667	0.035	18
6	0.43	0.39	0.066666667	0.146666667	0.056666667	0.056666667	0.065	18

No próximo passo iremos somar as diferenças de cada segmento para extrairmos um total que será utilizado para encontrarmos o segmento que mais se assemelha a nossa nova entrada.

Etapas 5, soma (calculando o total de diferenças):

1	0.7	18
2	2.493333333	19
3	4.906666667	19
4	2.575	19
5	1.093333333	18
6	1.211666667	18

Observe que a menor diferença total pertence ao segmento de ID 1, logo iremos retornar como resposta o output do ID de número 1 que é 18. E o nosso segmento de entrada ficará assim:

2.41	3.50	6.99	8.99	11.02	14.21	16.15	18
------	------	------	------	-------	-------	-------	----

E assim por diante... dessa forma conseguimos computar os dados exatamente como eles são computados de forma lógica pelo cérebro. A única limitação está em simular a variação decimal das oscilações quânticas. Aqui estamos utilizando qualquer precisão decimal aleatória de exemplo, infelizmente em um computador convencional igual aos que você tem na sua casa nós também teríamos que utilizar valores aleatórios gerados por uma função de randomização numérica para simular os erros da imprecisão neuronal. Isso no passado, atualmente já existem computadores quânticos que pelo fato de operarem a nível quântico conseguem emitir resultados aleatórios muito próximos dos que seriam obtidos em processos neuronais. Para o Físico Richard Feynman que elaborou a primeira proposta de um computador quântico em 1982, um computador que computasse a nível quântico teria melhores resultados na simulação de fenômenos quânticos reais da natureza. Com esse intuito foram criados os primeiros algoritmos quânticos de aprendizado de máquina com foco em fenômenos naturais que envolvessem interações materiais a nível quântico como por exemplo interação entre moléculas, colisões em aceleradores de partículas, compostos químicos e sinapses neuronais.

Graças a esses erros decimais quando o seu cérebro envia um sinal de output para que a sua perna dê um passo após o outro os passos nunca terão a mesma medida exata de um para o outro, se você saltar o seu próximo salto jamais terá a mesma altitude exata do salto anterior e por mais que você não perceba a sua voz nunca diz as palavras com a mesma entonação e assim sucessivamente para qualquer ação, pensamento ou reação do seu corpo.

Simular eventos naturais com 100% de precisão ainda é impossível mesmo para os computadores quânticos atuais que ainda possuem poucos bits quânticos, mas se isso fosse possível, um dia se tivéssemos todas as variáveis poderíamos utilizar o determinismo do universo para ter a capacidade de prever o que uma pessoa fará no dia seguinte e nos

próximos dias da sua vida até a sua morte, inclusive poderíamos prever o dia exato da sua morte e do que ela iria morrer com a hora exata do ocorrido. Também poderíamos utilizar o mesmo método para descrever eventos do passado que para nós são desconhecidos com 100% de precisão e revelar os segredos mais obscuros do universo. Mas é claro que por enquanto isso só é possível na imaginação da ficção científica, mesmo assim não há nenhuma lei da Física que nos impediria de fazer se nós tivéssemos as condições para isso.

Também é possível que os segmentos de memória retornem segmentos de saída no lugar de uma única variável de resposta quando há uma forte correlação entre as saídas de um ou mais segmentos. Por exemplo, quando estamos aprendendo a andar toda vez que a sua mente salva o segmento de um passo ela também salva o segmento do outro para completar o movimento ao se deslocar do ponto A ao ponto B. Ao encontrar uma forte correlação entre eles a sua mente une esses segmentos em um só. Nesse caso seria reconhecido um padrão de movimento que para o comando “andar” será sempre necessário movimentar uma perna e depois a outra. Então o resultado seria algo parecido com isso:

Variável 1	Variável 2	Variável N	Output 1: MOVIMENTAR A 1ª PERNA	Output 2: MOVIMENTAR A 2ª PERNA
------------	------------	------------	---------------------------------	---------------------------------

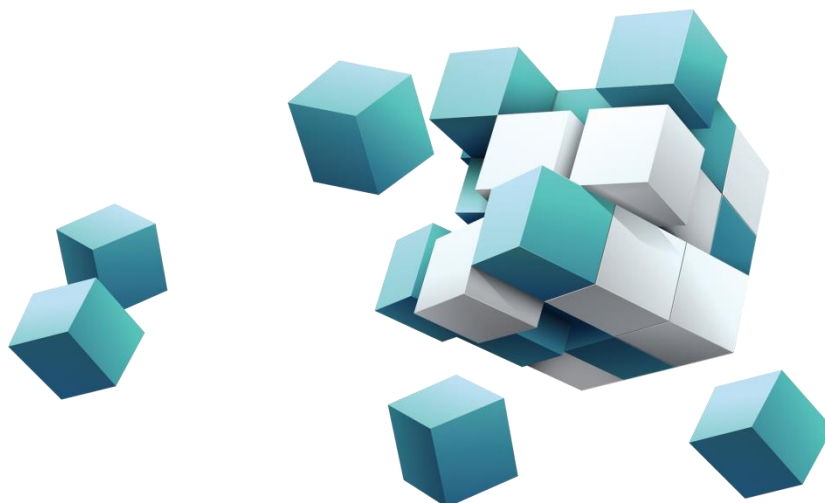
Observe que neste caso seriam retornadas duas variáveis de uma só vez que seriam interpretadas sequencialmente, primeiro o output 1 e depois o output 2.

Nos exemplos anteriores mantivemos os valores de saída como números inteiros, mas as flutuações quânticas também incidem sobre eles, o que nos faz ter pensamentos e executar movimentos ligeiramente ou até drasticamente diferentes uns dos outros mesmo quando a nossa intenção for repetir o evento anterior. Essas variações podem ocorrer na memorização das variáveis, na computação delas e/ou na interpretação das mesmas como valores de saída.

Substituindo o movimento da primeira perna pelo número 1 e o movimento da segunda perna pelo número 2, teríamos algo semelhante a isso:

Variável 1	Variável 2	Variável N	Output 1: 1.25369817	Output 2: 2.51497208
------------	------------	------------	----------------------	----------------------

Lembrando que a precisão decimal tem caráter estocástico, logo seus valores são puramente randômicos. Outro ponto importante a se destacar é que as variáveis lógicas da mente são computadas em espaços virtuais de 3 dimensões, mas aqui para facilitar estamos utilizando sempre segmentos de memória vetoriais em 2 dimensões. A organização virtual desses blocos de memória na mente assumiria na realidade um aspecto lógico semelhante ao da figura abaixo:



Os blocos de memória mais claros representariam as variáveis de saída e os mais escuros as variáveis de entrada, enquanto toda a estrutura seria a representação aproximada de um segmento imaginário completo. **É importante destacar que essa representação de variáveis em um segmento 3D é uma abstração lógica da parte virtual da mente e não corresponde ao seu aspecto físico real.**

Os segmentos de memória responsáveis pela geração e processamento da mente que vimos até aqui estão distribuídos por todo o cérebro e até mesmo por algumas poucas regiões do corpo. Mas a maior parte das informações, principalmente as mais importantes que envolvem raciocínio e cognição estão concentradas massivamente no cérebro. Existem alguns casos que ainda estão sendo estudados de pacientes que receberam órgãos transplantados e passaram a desenvolver gostos, comportamentos e/ou atitudes e até mesmo sentimentos muito específicos que pertenciam ao doador, mostrando que de fato a mente está atrelada a matéria e não pode existir sem ela. Alguns estudiosos acreditam que isso ocorra devido aos segmentos de memória armazenados em neurônios do órgão doador. Esse tipo de fenômeno costuma ser mais recorrente em transplantes de coração, mas também vem sendo observado em diversos tipos de transplantes de órgãos.

Os segmentos de memória apesar de gerados, processados e interpretados de forma elétrica em representações digitais, são armazenados de forma majoritariamente analógica através da estrutura morfológica das conexões neuronais. Os neurônios possuem uma variedade que tende ao infinito de combinações entre o número de terminais de entrada e saída que os conectam uns aos outros. Cada tipo de morfologia neuronal gera uma combinação elétrica diferente com uma carga muito específica de cada neurônio dependendo dessa combinação entre o número de inputs e o número de outputs formados pela estrutura física desses terminais.

Essas ramificações que compõe os terminais de entrada e saída são geradas através de uma espécie de “cicatriz” celular que acontece quando a célula neuronal se recupera rapidamente da descarga elétrica gerada pela informação recebida pelos nervos sensoriais que estão por

todo o corpo. É por isso que na psiquiatria chamados os traumas psicológicos de traumas, por que certos tipos de eventos podem emitir uma descarga elétrica tão forte de informação que essas “cicatrices” (traumas) celulares criam uma quantidade acima do normal de terminais de entrada e saída nos neurônios responsáveis pelo armazenamento dessa informação fazendo com que as variáveis tenham um peso que normalmente não teriam.

Como a informação é armazenada de forma analógica utilizando a própria estrutura física do cérebro isso nos impede de esquecermos as coisas intencionalmente, por que como o armazenamento não é digital não há como simplesmente apagar um segmento de memória a menos que ocorra uma lesão cerebral que faça com que o indivíduo perca fisicamente a parte do cérebro responsável por armazenar este segmento.

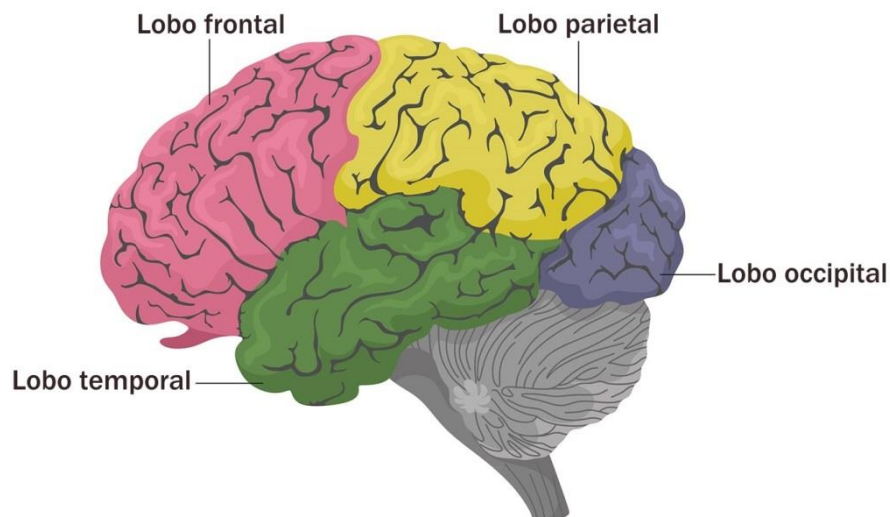
Na verdade quando achamos que esquecemos algo nós não esquecemos, pois uma vez que a informação foi recebida à morfologia do cérebro jamais voltará a ser o que ela era antes. As coisas que achamos esquecer na verdade só foram encobertas por ramificações neuronais maiores que surgiram a sua volta, mas essa informação tida como “esquecida” teoricamente ainda estaria lá e poderia surgir do “nada” simplesmente quando essas ramificações que a “soterraram” se tornassem mais fracas ao decorrer do tempo.

Geralmente o que os tratamentos psicológicos e psiquiátricos fazem não é apagar as variáveis traumáticas, mas inserir novas variáveis que sejam capazes de se sobressaírem sobre os traumas a fim de controlá-los.

Quando ocorre uma lesão em uma ou mais regiões do cérebro as variáveis que foram salvas na respectiva região também serão comprometidas, podendo virem a serem corrompidas o que acarretaria na geração de outputs muito imprecisos ou até mesmo deletadas se a parte física que as armazenava for destruída. Neste caso se a lesão ocorrer de maneira muito súbita à informação será apagada imediatamente com perda da região responsável pelo armazenamento, mas se a lesão ocorrer de maneira menos súbita ou gradativa o cérebro terá tempo de fazer o “backup” dessas memórias as realocando para outras regiões podendo vir até mesmo em alguns casos utilizar regiões responsáveis por armazenarem variáveis de funções completamente diferentes, fazendo com que a região receptora desenvolva habilidades que normalmente ela não desenvolveria.

Partindo de uma visão da neurociência computacional a morte cerebral também conhecida como morte encefálica acontece quando as variáveis responsáveis por manterem a mente consciente são deletadas, mantendo somente as variáveis responsáveis por algumas das funções fisiológicas, seria um corpo sem mente. Como neste caso há uma destruição da morfologia responsável pelo armazenamento dessas variáveis o processo é irreversível. Já a morte efetiva acontece quando todas as variáveis são deletas.

Confira na figura abaixo os principais clusters/grupos de variáveis e suas respectivas regiões de armazenamento no cérebro:



Lobo frontal: armazena as variáveis responsáveis pela personalidade, tomada de decisão e movimento.

Lobo parietal: armazena as variáveis responsáveis pela identificação de objetos, relações espaciais, dor, tato e língua falada.

Lobo occipital: armazena as variáveis responsáveis pela visão.

Lobo temporal: armazena as memórias de curto prazo, funciona como a memória RAM do cérebro além de armazenar as variáveis da fala, do ritmo musical e do olfato.

Os demais tipos de variáveis não possuem regiões específicas e podem ser armazenadas por toda a área cerebral.

Se por exemplo à região do Lobo frontal for lesionada uma pessoa poderá ter desde alterações sutis de personalidade até alterações drásticas de comportamento ou se por acaso o Lobo occipital apresentar alguma lesão isso poderá comprometer efetivamente a visão e assim sucessivamente para todas as regiões. O grau de comprometimento das funções irá variar proporcionalmente com o grau de comprometimento da região afetada fisicamente.

É muito comum, por exemplo, lutadores com o passar da idade apresentarem alterações de personalidade, terem sua capacidade de decisão comprometida e/ou lentidão ao se movimentar para mover os membros ou os músculos da fala por receberem muitos impactos na região do Lobo frontal que é responsável por armazenar as variáveis dessas funções.

A representação lógica da clusterização/agrupamento de segmentos de memória virtual no cérebro poderia ser representada da seguinte forma usando x para as variáveis de entrada e y para as saídas nos índices/identificadores de inserção i:

Lobo frontal

i1	x1	x2	xn	y1	yn
i2	x1	x2	xn	y1	yn
in	x1	x2	xn	y1	yn

Lobo parietal

i1	x1	x2	xn	y1	yn
i2	x1	x2	xn	y1	yn
in	x1	x2	xn	y1	yn

Lobo temporal

i1	x1	x2	xn	y1	yn
i2	x1	x2	xn	y1	yn
in	x1	x2	xn	y1	yn

Lobo occipital

i1	x1	x2	xn	y1	yn
i2	x1	x2	xn	y1	yn
in	x1	x2	xn	y1	yn

Dessa forma quando o segmento de entrada para a predição for recebido por um estímulo sensorial do Lobo frontal a computação das variáveis será enviada para o cluster do Lobo frontal, a mesma coisa aconteceria para estímulos controlados pelo Lobo parietal, temporal e occipital fazendo com que o cérebro realize computação paralela dividida em 4 núcleos principais de processamento.

Aprendizado de Máquina

Com base no funcionamento da lógica matemática da mente derivou-se uma subárea de aplicação da computação neurocientífica que mais tarde faria parte da inteligência artificial que é o aprendizado de máquina. O aprendizado de máquina consiste em um conjunto de técnicas matemáticas inspiradas na assimilação e aplicação do conhecimento computado pela mente humana que podem ser traduzidas em algoritmos computacionais.

Esses algoritmos além de utilizados na neurociência também podem ser aplicados na estatística onde por muito tempo foram conhecidos como algoritmos de estatística multivariada antes da popularização do termo aprendizado de máquina. Além da estatística esses algoritmos possuem diversas aplicações matemáticas desde a econometria, física, química, biologia e mais recentemente na computação quântica dentre diversos outros campos de estudo que fazem uso da matemática aplicada.

Os algoritmos de aprendizagem de máquina ou machine learning como são conhecidos internacionalmente foram elaborados com base em abstrações matemáticas de um ou mais aspectos da matemática da mente. Eles utilizam duas fases principais durante o cálculo que são a fase do reconhecimento de padrões (**treinamento**) onde serão passados vetores de amostra com valores de entrada e suas possíveis saídas representando os segmentos de memória da mente e uma fase de predição. No treinamento assim como ocorre no treino de determinadas tarefas para que seres humanos possam assimilar conteúdo, esses dados de entrada serão varridos em busca de padrões que possam ser utilizados para explicar as saídas definidas para cada vetor de entrada. Quando esses padrões forem assimilados o treinamento não será mais necessário nas próximas vezes que dados diferentes com os mesmos padrões forem passados sem o valor de resposta para a segunda fase que é a fase de execução/aplicação dos padrões (**predição**). Na fase de predição serão recebidos vetores com variáveis que possuam o mesmo padrão de distribuição numérica dos dados usados no treinamento, com a diferença de que agora os valores de saída serão desconhecidos e deverão ser adivinhados no resultado da predição através da aplicação dos padrões que foram reconhecidos no treinamento.

Esse reconhecimento e aplicação de padrões possibilita que o usuário possa prever cenários futuros com base em cenários de históricos passados, utilizando apenas os valores de entrada como parâmetro da predição. Isso da ao desenvolvedor à capacidade de prever esses cenários com resultados exatos ou próximos da exatidão.

Da mesma forma que um ser humano quando nasce aprende com eventos e tarefas que contenham resultados de exemplo como: **um copo cai (input) e quando cai quebra (output), dois mais dois (input) quando somados se tornam quatro (output), o fogo (input) queima (output), quando sinto fome (input) devo comer (output) e etc...** Os algoritmos de aprendizado de máquina também utilizam exemplos de saída para as entradas do treinamento com o objetivo de aprender. Uma vez que o conhecimento é aprendido, basta aplica-lo a qualquer entrada da predição para que a saída correspondente seja prevista com base na saída do treinamento que possua valores de entrada semelhantes à entrada da predição.

Para o nosso exemplo de aprendizado de máquina iremos utilizar o algoritmo **K-Nearest Neighbors** (K-enésimo Vizinho mais Próximo) ou KNN. Nesse algoritmo alimentamos o treinamento com uma ou mais variáveis de entrada que possuam exemplos de saída e para calcular o resultado da predição fazemos a diferenciação entre a entrada da predição e as entradas do treinamento. Para isso iremos aplicar um cálculo de distância euclidiana subtraindo a entrada da predição de cada entrada do treinamento. Um método muito parecido com o da matemática mental, com a principal diferença de que a mente computa em 3D e no KNN iremos computar em 2D. Vejamos como ficaria no exemplo abaixo...

Entradas e saídas do treinamento:

TREINAMENTO		
ID	Idade em anos (input)	Fase da vida (output)
1	10	criança
2	15	adolescente
3	30	adulto
4	70	idoso

Entrada da predição:

PREDIÇÃO	
	Idade em anos (input)
	14

Cálculo da diferenciação com base na distância euclidiana:

Subtração	
ID	Predição - Treinamento
1	-4
2	1
3	16
4	56

Exponenciação ao quadrado para positivar números negativos:

Exponenciação	
ID	Subtração ²
1	16
2	1
3	256
4	3136

Radiciação quadrada para voltar os números à proporção original:

Radiciação	
ID	$\sqrt{\text{Exponenciação}}$
1	4
2	1
3	16
4	56

Se estivéssemos utilizando mais de uma variável por entrada nós somaríamos as raízes de cada linha reduzindo cada entrada em um único valor. Neste caso como temos apenas uma única variável de entrada por linha isso não será necessário.

Agora basta definir o valor de K para obtermos a nossa resposta. O valor do parâmetro K no KNN refere-se à quantidade de vizinhos mais próximos da menor diferença, dentro desses vizinhos mais próximos contamos as ocorrências de cada uma das saídas e retornamos como resposta a saída com a maior frequência entre os vizinhos selecionados. Aqui como temos uma saída diferente para cada linha iremos utilizar 0 para K (sem vizinhos). Se estivéssemos usando K=2 por exemplo, teríamos como vizinhos de 1 (a menor diferença) os números 4 e 16, depois retornaríamos como output de resposta a saída que mais se repete entre as saídas das linhas 1, 2, 3 correspondentes aos resultados 1, 4 e 16. Os vizinhos serão sempre selecionados do mais próximo até o menos próximo da menor diferença.

Como estamos utilizando K=0 teremos como resposta a saída da menor diferença que no nosso caso será a saída da linha 2 que possui o output de resposta igual a adolescente.

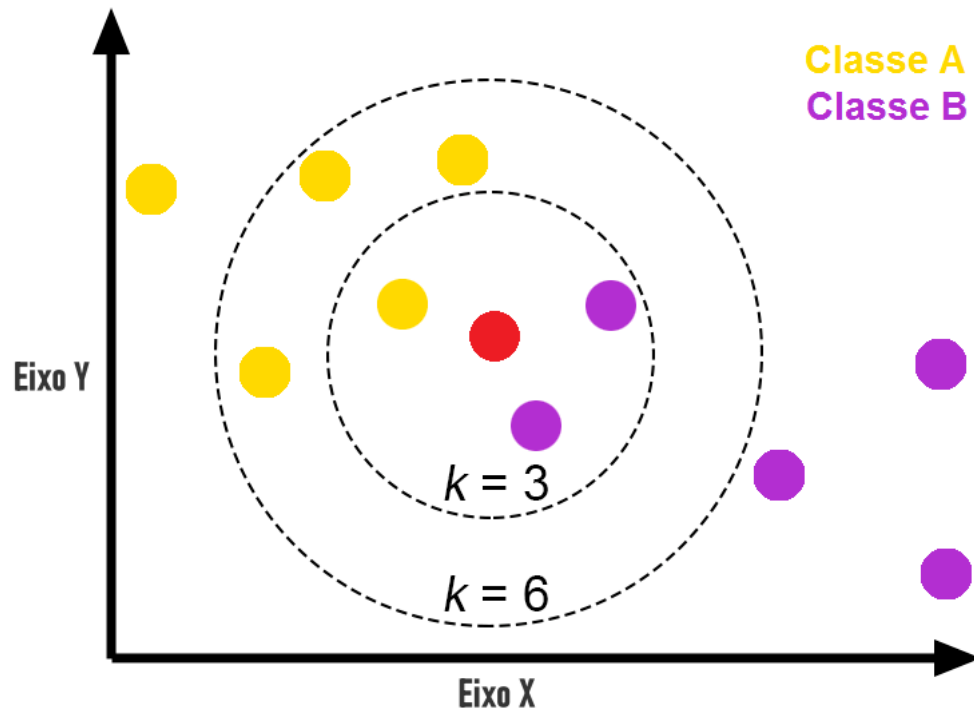
Radiciação		
ID	$\sqrt{\text{Exponenciação}}$	output
1	4	criança
2	1	adolescente
3	16	adulto
4	56	idoso

Resposta da predição: adolescente. Logo um indivíduo de 14 anos com base nos dados de treinamento corresponde a um adolescente.

PREDIÇÃO	
Idade em anos (input)	Fase da vida (output)
14	adolescente

Observe que conseguimos prever um valor desconhecido apenas utilizando alguns valores conhecidos que foram passados para o treinamento. **A quantidade de exemplos necessários no treinamento irá depender da complexidade do seu conjunto de dados.** Aqui como usamos no nosso exemplo um caso de classificação simples, uma entrada para cada saída possível já foi o suficiente.

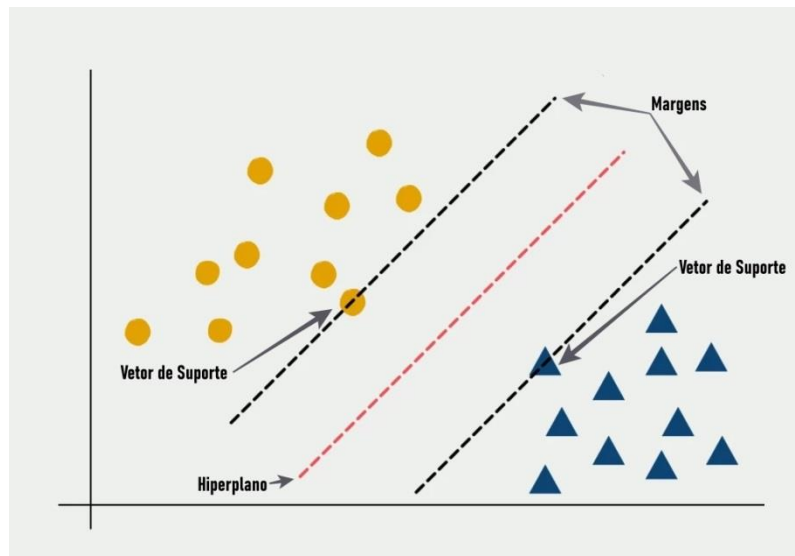
Supondo que os nossos dados de treinamento tivessem apenas dois tipos de output, sendo eles a Classe A e a Classe B, ao distribuírmos as entradas de treinamento em um plano cartesiano teríamos algo semelhante a isso:



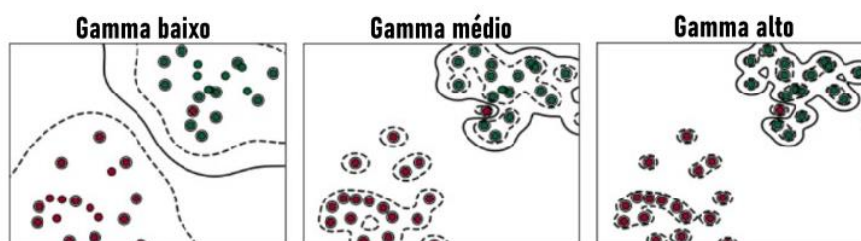
Os pontos fictícios em amarelo representam as entradas da Classe A, os pontos roxos as entradas da Classe B e o ponto vermelho ao centro seria a entrada da predição. Note que para $K=3$ teríamos a verificação da classe que mais se repete nas três entradas mais próximas do dado que queremos prever. Como para $K=3$ dentre as três entradas do treinamento mais próximas selecionadas a classe que mais se repete é a B (pontos roxos), a nossa resposta seria a Classe B. No entanto se estivéssemos utilizando $K=6$ o nosso resultado seria completamente diferente já que os pontos que mais se repetiriam seriam os amarelos fazendo com que a nossa resposta da predição neste caso fosse a Classe A.

Outro algoritmo de aprendizado de máquina bastante utilizado é o **Support Vector Machine** (Máquina de Vetores de Suporte/Máquina de Suporte Vetorial) ou SVM. Esse algoritmo usa as entradas de uma classe que estejam mais próximas das entradas da outra para separar os dados, como as entradas normalmente vêm em seguimentos de memória (vetores), essas entradas opostas mais próximas recebem o nome de vetores de suporte, o termo máquina vem de aprendizado de máquina. Ao encontrar os vetores de suporte será traçada uma linha para cada classe que passe sobre eles de modo a determinar margens que denotem o limite das classes envolvidas onde cada classe terá a sua área específica. Depois entre uma margem e outra será traçada outra linha que passe exatamente no meio delas, essa última linha é chamada de hiperplano e qualquer entrada da predição que for posicionada do lado do

hiperplano que represente uma determinada classe terá como output de resposta a classe pertencente a essa área.

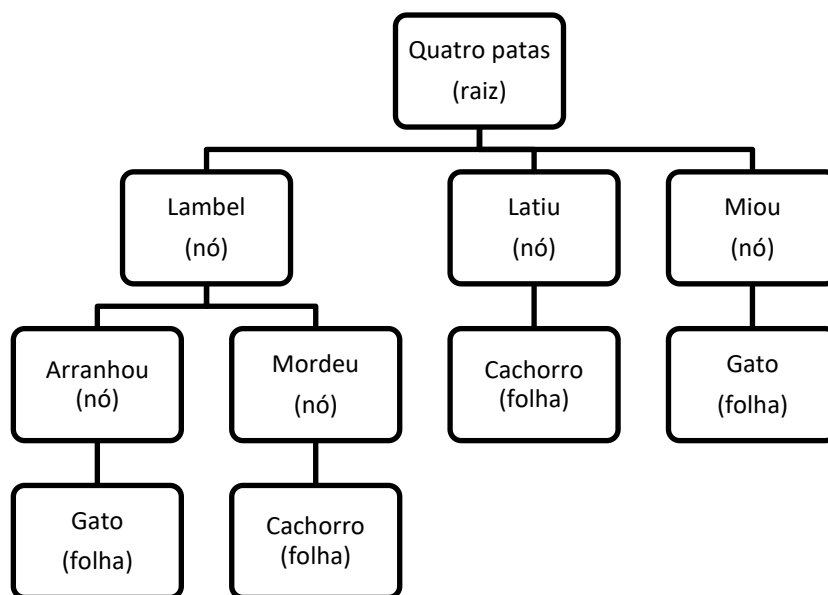


No exemplo acima temos duas classes, a classe amarela representada por círculos e a classe azul representada por triângulos. Os vetores de suporte são as entradas opostas do treinamento que mais se aproximam entre si, sobre elas temos as linhas tracejadas de preto representando as margens de limite entre as classes e exatamente no meio entre elas temos o hiperplano tracejado de vermelho. Qualquer entrada da predição que seja representada do lado esquerdo do hiperplano será classificada com a classe amarela e qualquer entrada que seja representada a direita do hiperplano será classificada como azul. Também é possível utilizar um parâmetro matemático de nome “gamma” que irá controlar o nível de separação entre as classes, quanto maior o gamma, maior será a separação, ou seja, quanto maior o gamma menor será a área envolvendo as classes.



Note na imagem acima que a área de ocupação das classes tende a se justificar conforme o valor do parâmetro gamma aumenta. Neste caso temos uma classe de pontos vermelhos e outra de pontos verdes, o hiperplano se aproxima de uma das classes (neste exemplo da verde) à medida que a área de ocupação delas diminui. O hiperplano está representado pela linha contínua e as margens pelas linhas tracejadas. Como aqui o hiperplano se fecha a partir da segunda imagem, qualquer coisa dentro do hiperplano receberá a classificação verde e qualquer coisa fora a classificação vermelha.

Outro algoritmo baseado em processos mentais que é bastante utilizado é o de Árvore de Decisão ou “**Decision Tree**”. Nesse algoritmo chegamos ao resultado final comparando as alternativas possíveis de forma categórica em uma estrutura em forma de uma árvore de cabeça para baixo. Por exemplo, se o animal latir eu não preciso conferir as outras características para constatar que se trata de um cachorro. Porém se ele não latir, mas se tiver quatro patas, focinho longo e abanar o rabo é possível chegar ao mesmo resultado (cachorro) sem a variável “latir”.



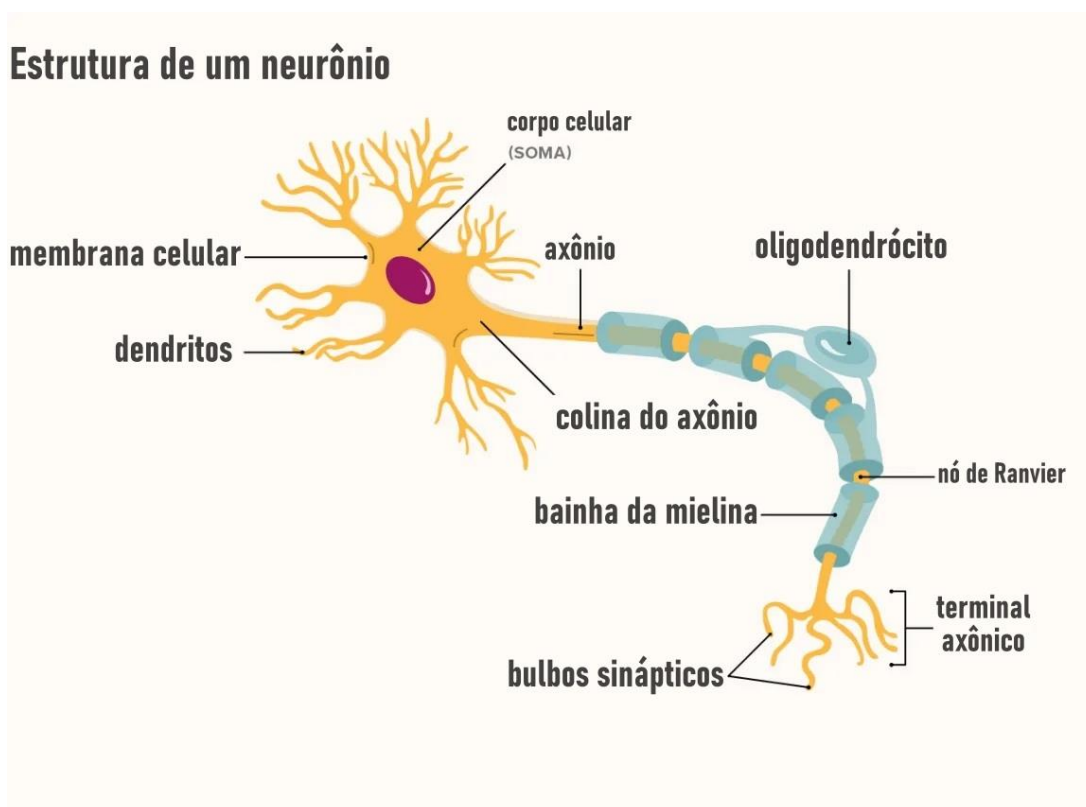
Nos algoritmos de Árvore de Decisão chamamos a alternativa principal no topo de **raiz** as alternativas secundárias de **nós** e as respostas de **folhas**. O conjunto composto por **nós** e **folhas** chamamos de **ramo**, no exemplo acima temos três **ramos**. A estrutura da árvore poderá conter dezenas, centenas, milhares ou até milhões de ramificações dependendo da complexidade da base de dados.

A Matemática do Cérebro

Se você chegou até aqui provavelmente já deve ter percebido que a mente computa informação de forma lógica interpretando sinais elétricos a nível digital, mas como foi mencionado em tópicos anteriores a parte digital da mente na computação dos dados é minoritária, pois o cérebro computa informação majoritariamente de forma analógica usando a sua própria morfologia para isso. Mas como essa computação analógica é feita? Será isso que veremos a partir de agora.

Como estamos tratando de neurociência computacional podemos traçar uma analogia de que a mente seria o software do cérebro, logo seria a parte digital do mesmo e toda a sua parte matemática seria computada digitalmente. Mas a mente depende do cérebro para existir e o cérebro é analógico, além disso, é o responsável por computar a maior parte da informação inclusive de forma inconsciente e automática, a parte consciente relativa à mente é apenas uma pequena parte desse sistema biológico.

Primeiramente iremos analisar a arquitetura biológica do neurônio que é a unidade fundamental de informação analógica no cérebro.



* **corpo celular:** O corpo celular, ou pericário, é a parte do neurônio onde se encontra o núcleo e o citoplasma que o envolve. (O citoplasma é uma região da célula localizada entre o núcleo e a membrana celular).

* **membrana celular:** A membrana celular, plasmalema, membrana plasmática ou membrana citoplasmática é uma fina película formada principalmente por proteínas que protege o citoplasma da célula.

* **axônio:** O axônio é o prolongamento único de uma célula nervosa, por onde é transmitido o estímulo nervoso recebido através do corpo celular em forma de sinal elétrico e retransmitido para um ou mais terminais de saída.

* **dendritos:** Os dendritos são prolongamentos do neurônio que recebem o estímulo nervoso dos terminais de outro neurônio e o transmite para o corpo celular.

* **oligodendrócito:** O oligodendrócito ou célula de Schwann é um tipo de célula acoplada aos neurônios que fornece suporte e isolamento para os axônios de um ou mais neurônios ao mesmo tempo. São encontrados somente nos neurônios do sistema nervoso central.

* **colina do axônio:** A colina do axônio ou outeirinho do axônio é a região que une o axônio a soma do corpo celular.

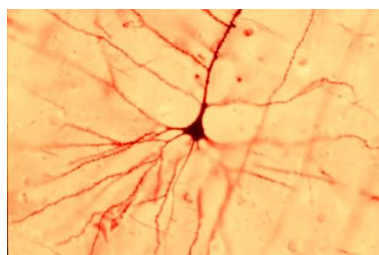
* **nó de Ranvier:** O nó de Ranvier ou nódulo de Ranvier é responsável por transmitir a informação através do canal do axônio.

* **bainha da mielina:** A bainha da mielina é uma estrutura celular que atua como isolante elétrico do axônio.

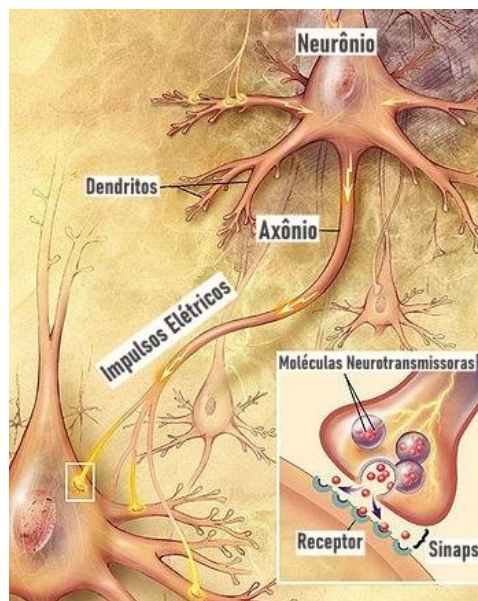
* **terminal axônico:** O terminal axônico ou terminal do axônio é uma ramificação nervosa que transmite sinais elétricos para os dendritos de outros neurônios.

* **bulbos sinápticos:** Os bulbos sinápticos são as estruturas mais próximas da fibra muscular encontradas no final dos terminais axônicos e responsáveis pela junção neuromuscular do sistema nervoso.

A informação é recebida em forma de sinais elétricos pelos dendritos que transferem a carga elétrica para o corpo celular protegido pela membrana celular, o núcleo emite um estímulo nervoso que contrai o corpo celular e transforma um ou mais sinais elétricos dos dendritos em uma carga única que é passada para o canal do axônio através da colina do axônio, a carga elétrica do axônio é isolada pela bainha da mielina e impulsionada pelos nós de Ranvier até os terminais axônicos de saída que liberam a eletricidade pelos bulbos sinápticos para que ela possa ser recebida pelos dendritos do próximo neurônio gerando assim uma reação em cadeia que só termina com a nossa morte. Confira a seguir uma imagem real aproximada por microscópio das redes neurais biológicas contidas no cérebro:



Durante muito tempo pensou-se que os terminais de saída de um neurônio se tocavam com os terminais de outro para transmitir a informação elétrica, porém hoje sabemos que não acontece toque entre as terminações nervosas, a transferência elétrica é feita a curta distância por um vão extremamente sutil, muito difícil de se observar até mesmo em microscópios, esse vão que há entre os terminais de saída e entrada quando acometido de eletricidade se transforma no que chamamos de sinapse. A descarga elétrica é minuciosamente calibrada através de moléculas neurotransmissoras contidas nas extremidades dos bulbos sinápticos. As partículas de elétron fluem momentaneamente em milionésimos de segundos do terminal de saída do neurônio corrente em direção aos receptores dos terminais de entrada (dendritos) do neurônio posterior mais próximo.

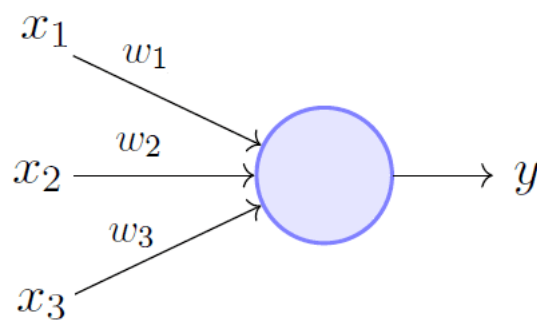


Agora que já vimos o funcionamento dos neurônios biológicos e como eles operam de forma analógica, como traduzir isso em informação matemática digital para aplicarmos o mesmo princípio nos computadores? Por muito tempo não tivemos essa resposta até que em 1958 o pesquisador Frank Rosenblatt cria o **Perceptron**, que nada mais é do que uma fórmula matemática para traduzir de forma aproximada os principais elementos neuronais envolvidos na computação de dados elétricos em linguagem matemática.

$$y = w_0 + \sum_{i=1}^n x_i w_i$$

Essa é a primeira versão do **Perceptron** onde para calcularmos uma saída “**y**” multiplicamos a entrada “**x**” por um peso aleatório “**w**”. Os resultados desses produtos das entradas pelos pesos serão somados e se necessário um valor de viés poderá ser acrescentado para calibrar o cálculo. Aqui o viés está sendo representado por “**w0**” mas também costuma ser representado por “**b**”, “**θ**” ou uma variável qualquer. A saída calculada é comparada com uma saída de exemplo e se o resultado encontrado estiver distante do desejado o calculo se repete de **1** a “**n**” vezes até que a resposta para “**y**” seja satisfatória. Quando o peso adequado for encontrado ele será anotado para que na predição esse processo repetitivo de **1** a “**n**” não seja mais necessário.

O **Perceptron** é o que chamamos na matemática de **função de aproximação**. “Dado um número o aproxime o máximo que puder de outro”. Confira na imagem a seguir uma representação gráfica simples da arquitetura matemática do **Perceptron**:



Na representação gráfica acima não estamos utilizando o valor de viés por não se tratar de uma variável obrigatória. O **Perceptron** considera somente os elementos principais da anatomia neuronal, no caso as entradas de “**x1**” até “**xn**” representam os dendritos, os pesos de “**w1**” até “**wn**” as cargas elétricas, a **função somatória** representaria a “soma” do corpo celular e o resultado “**y**” o sinal de resposta dos terminais do axônio.

Partindo do pressuposto matemático de que qualquer número pode ser representado pela multiplicação de dois outros o **Perceptron** de forma simples consegue reproduzir qualquer padrão numérico utilizando apenas a multiplicação de dois fatores (entrada e peso).

Mas se você for uma pessoa atenta já deve ter percebido que ficar calculando isso “**n**” vezes “chutando” pesos em uma folha de papel não é um processo viável. Foi por esse motivo que o **Perceptron** ficou sem uma aplicação prática até a chegada da era dos computadores. Com um computador isso poderia ser feito de forma 100% automatizada tornando o cálculo bem mais prático do que antes. E foi somente a poucos anos atrás com a popularização dos computadores e o aumento expressivo da capacidade computacional que o **Perceptron** ganhou a fama que tem hoje.

Com o tempo novas estruturas matemáticas foram sendo inseridas ao **Perceptron** com o objetivo de torná-lo mais rápido e preciso como, por exemplo, o método de **descida de gradiente** (ou **gradiente descendente**) que é aplicado na otimização dos pesos. Ele recebe esse nome por que ele faz com que a **taxa de erro** também conhecida como “**perda**” diminua de forma gradativa (**gradiente**) a cada nova iteração. Com o **método do gradiente** já não era mais necessário ficar “chutando” os pesos até encontrar um valor próximo do esperado, agora podemos fazer isso de forma consciente tornando o processamento muito mais rápido.

Existem várias maneiras de se calcular o **gradiente**, o primeiro método utilizado por exemplo foi o de subtrair o valor da saída esperada para “**y**” do valor calculado de “**y**” e se esse valor for positivo aumentamos gradativamente o valor dos pesos caso contrário diminuimos. Isso seria feito repetidas vezes em um processo chamado de **backpropagation** (ou **retropropagação**). O **backpropagation** é a repetição do cálculo desde o início utilizando novos pesos diferentes dos anteriores. O número total de **backpropagations** é definido por uma variável chamada de “**épocas**” (também conhecida pelo termo inglês “**epochs**”) que delimita o limite de repetições do **backpropagation** para que o algoritmo não caia em um loop infinito.

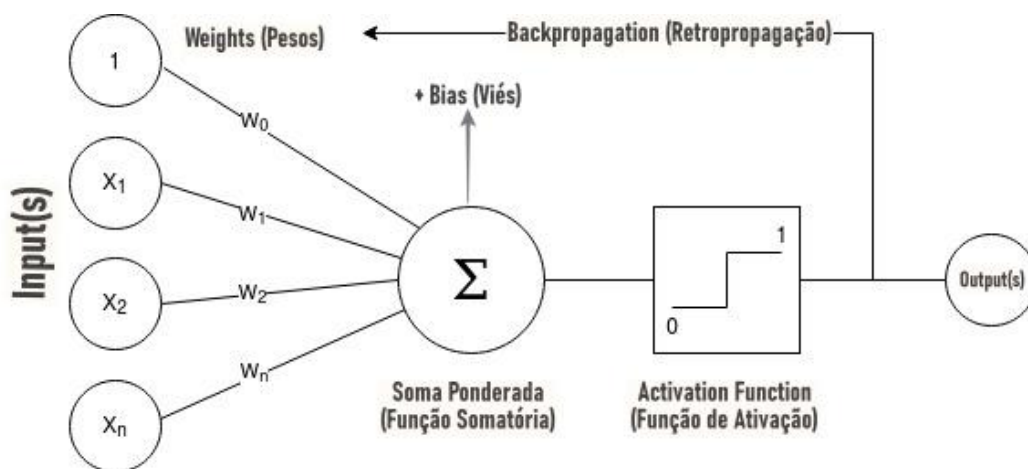
Outro método que foi inserido ao **Perceptron** com o passar do tempo foi o de utilizar uma função antes de emitir o resultado com o objetivo de formatar o valor de “**y**”. Essa função ficou conhecida como **função de ativação** e simula o axônio responsável por ativar os terminais de saída eletricamente. Como o axônio de certa maneira formata a carga elétrica com uma distribuição diferente da recebida pelo corpo celular essa função tenta copiar esse comportamento com o objetivo de personalizar a escala dos valores de resposta. Isso pode ser útil em casos de classificação onde só podemos emitir resultados dentro de um número limitado de categorias.

O que diferencia matematicamente o **neurônio artificial (Perceptron)** do neurônio natural (biológico) é que como o neurônio natural é analógico ele se torna lento ao computar dados matemáticos, mas é extremamente bem sucedido ao computar dados cognitivos. Por isso somos tão bons em reconhecer sensações e sentimentos ao mesmo tempo em que somos lentos ao fazer contas muito complexas já que números são melhor representados digitalmente. Já no neurônio artificial acontece exatamente o contrário, por serem estruturas matemáticas puramente digitais se tornam muito mais eficientes do que nós em reconhecer padrões numéricos, mas ainda muito limitados no reconhecimento de padrões cognitivos.

A principal revolução causada pelo **neurônio artificial** foi que diferente dos algoritmos anteriores de aprendizado de máquina que nós vimos baseados somente na estrutura lógica do pensamento que atuam exclusivamente na classificação de dados categóricos, o **Perceptron** além de classificar as entradas retornando uma das saídas do treinamento também é capaz de retornar resultados adaptativos gerando saídas que não constam no histórico se a proporção das entradas da previsão for diferente da encontrada nas entradas do treinamento, na estatística chamamos esse tipo de cálculo de regressão. Ou seja, os **neurônios artificiais** são algoritmos extremamente flexíveis assim como o nosso cérebro podendo reconhecer tanto padrões classificativos quanto padrões regressivos. Isso muda tudo, pois agora não precisamos mais de um algoritmo específico para cada caso, podemos utilizar o mesmo algoritmo sem a necessidade de um novo treinamento com uma estrutura de cálculo e

configuração diferente para cada caso. Podemos reconhecer qualquer tipo de padrão usando somente o algoritmo de **neurônio artificial**.

O **Perceptron** atualmente trabalha com as entradas (ou camada de entrada) multiplicadas pelos pesos onde os produtos serão somados e poderão ou não ter um valor arbitrário de viés adicionado ao resultado da soma que em seguida passa por uma função de formatação (**função de ativação**) geralmente uma função “**sigmoide**” (com resultados entre 0 e 1) e envia esse resultado para a saída (ou camada de saída), se o resultado da saída não for satisfatório se inicia o **backpropagation** que ajustará os pesos através do **método de gradiente** até que o valor do resultado esteja próximo o suficiente da saída esperada.



Considere o seguinte exemplo para os padrões dos dados de entrada e saída da fase de treinamento:

Treinamento		
Input 1	Input 2	Output
2	3	10
2	5	14
4	7	22
Predição		
1	3	???

Poderíamos estar utilizando qualquer padrão de distribuição numérica para o nosso exemplo, mas neste caso observe que o padrão para se obter a saída consiste em somar as entradas e multiplicar o resultado da soma por dois, logo a saída para a predição das entradas **1** e **3** deveria ser **(1 + 3) x 2 = 8**. Porém se aplicarmos um dos algoritmos de aprendizado de máquina baseados na matemática da mente que vimos em tópicos anteriores como o **KNN** por exemplo teríamos **10** como resultado, isso por que as entradas **2** e **3** são as mais próximas de **1** e **3**. Agora vejamos como ficaria o resultado aplicando o **Perceptron**...

Para efeitos didáticos iremos pressupor que todos os **backpropagations** necessários já foram executados e obtivemos os pesos 2.5 e 1.71.

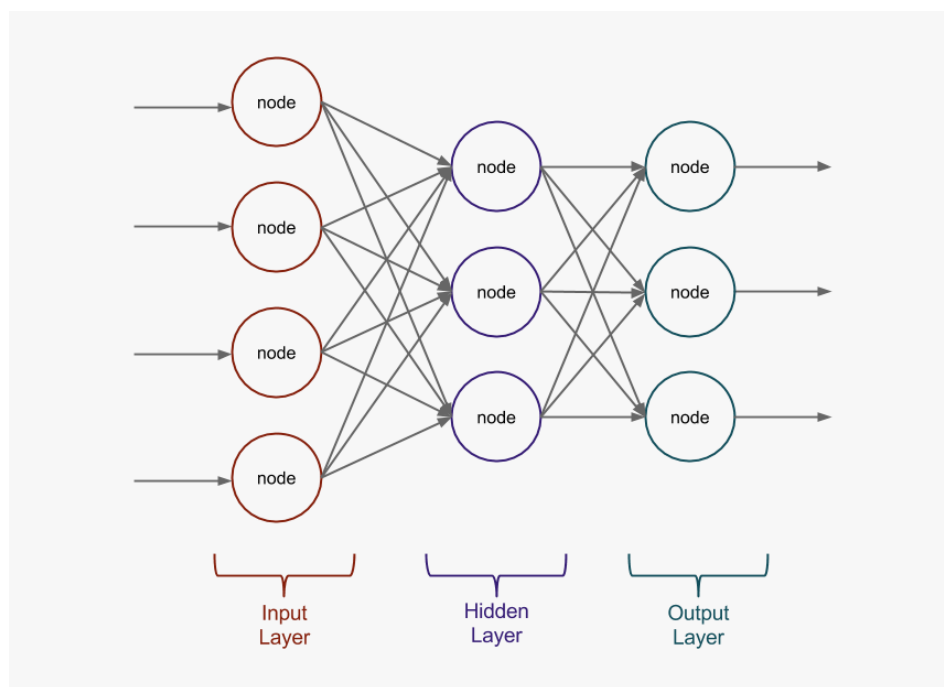
Cálculo do Perceptron		
Treinamento		
Input 1	Input 2	Output
$2 \times 2.5 = 5$	$3 \times 1.71 = 5.13$	$5 + 5.13 = 10.13$
$2 \times 2.5 = 5$	$5 \times 1.71 = 8.55$	$5 + 8.55 = 13.55$
$4 \times 2.5 = 10$	$7 \times 1.71 = 11.97$	$10 + 11.97 = 21.97$
Predição		
$1 \times 2.5 = 2.5$	$3 \times 1.71 = 5.13$	$2.5 + 5.13 = 7.63$

No exemplo acima estamos aplicando o cálculo novamente no treinamento apenas para que você possa conferir os valores, mas na vida real isso não seria necessário, para encontrar a resposta basta multiplicar as entradas da predição pelos pesos e somar os produtos, aqui como estamos em um caso de regressão não iremos formatar a saída dentro de um intervalo, ao invés disso iremos deixar o resultado livre para se adaptar as novas entradas desconhecidas.

Resultados 100% precisos dificilmente irão acontecer uma vez que o **neurônio artificial** segue a estrutura do **neurônio biológico** que computa com erros de predição ocasionados pelas flutuações quânticas da sinapse, neste caso o **Perceptron** simula esses erros de precisão emitindo resultados aproximados com precisão decimal, mas isso não quer dizer que a resposta esteja errada por que se arredondarmos os números chegaremos aos valores que estamos esperando. **Sempre que os valores de saída forem arredondados para a mesma quantidade de casas decimais dos dados de treinamento e isso ocasionar em resultados iguais ou semelhantes aos resultados esperados, poderemos considerar a saída como estando correta.** Obviamente como não estamos utilizando um computador quântico à parte decimal será apenas uma aproximação randômica do que teríamos nos erros de computação biológica, para obtermos erros realmente realistas com base no comportamento cerebral teríamos que executar o código em um computador quântico o que não é o nosso caso.

Com o tempo novos estudos foram surgindo e o **Perceptron** evoluiu para cálculos onde podemos conectar um **neurônio artificial** a outro nos aproximando ainda mais da estrutura analógica real do cérebro. Existem vários tipos de arquitetura de conexão onde podemos, por exemplo, transferir as saídas de um Perceptron para as entradas de outro ou simplesmente reiniciar o cálculo com as mesmas entradas usando pesos diferentes como explicamos anteriormente. A novidade é que agora podemos configurar uma ou mais **camadas** com um ou mais **neurônios** entre as camadas de entrada ("**input layer**") e saída ("**output layer**"), essas **camadas** extras recebem o nome de **camada oculta** ou "**hidden layer**". As conexões desses vários **neurônios** darão origem a um outro tipo de estrutura muito mais precisa e eficiente conhecida como **rede neural artificial**. As **redes neurais artificiais** são baseadas nas conexões entre dois ou mais **Perceptrons Simples** como os que vimos anteriormente dando origem a um

novo tipo de **Perceptron** chamado de **Multilayer Perceptron** que nada mais é do que o **Perceptron Simples** com uma ou mais **camadas ocultas**. Os pontos de conexão entre esses **neurônios** do **Multilayer Perceptron** são chamados de **nós**, (ou **nódulos**), a quantidade de nós em uma **camada** poderá ser definida pelo próprio programador que irá testar o treinamento com diversos tipos de configurações diferentes e escolher a configuração que retorne o melhor resultado. Geralmente em algoritmos de **redes neurais artificiais** a quantidade de nós é configurada através de um parâmetro de nome “**nodes**”. Confira a seguir como seria a estrutura gráfica de um **Multilayer Perceptron** com uma quantidade arbitrária de **camadas** e **nódulos** de exemplo...



No exemplo acima estamos usando **camadas de conexão densa** onde cada **nó** da camada corrente que corresponde à saída do **neurônio** atual é multiplicado por cada um dos **pesos** da camada seguinte, mas também existem **camadas de conexão esparsa** onde há um único **nó** para cada saída anterior em que cada **nó** da camada corrente está conectado a apenas um **nó** da próxima camada, dessa forma conectando apenas **nós** que estejam na mesma posição vetorial.

Qual a necessidade de se usar mais camadas? O número de camadas está diretamente ligado à quantidade de padrões que poderão ser reconhecidos. No reconhecimento de padrões em imagens onde temos que analisar aspectos como bordas, cores e proporções você precisará de muito mais camadas do que precisaria se estivesse aplicando a rede em dados tabulares por exemplo. Algumas arquiteturas de rede ainda podem utilizar um parâmetro de configuração conhecido como **taxa de aprendizagem** que é responsável por controlar o nível de assimilação desses padrões, pode ser que em conjuntos de dados muito complexos com diversos tipos de padrões você não queira que a sua rede assimile todos se os padrões que o interessa forem

apenas uma pequena parte dos que constam nos dados, nesse caso para não retornar resultados incorretos com base em padrões irrelevantes o valor percentual da **taxa de aprendizagem** poderá ser diminuído. É importante ressaltar também que você nunca deverá usar camadas além do necessário ou isso poderá deixar a sua fase de treinamento excessivamente lenta além de retornar resultados irreais com base em padrões muito sutis que não deveriam ser usados para influenciar o resultado da predição.

Não existe uma fórmula perfeita para se encontrar a melhor configuração para cada tipo de caso, o programador deverá fazer testes com diversos tipos de combinações configuratórias diferentes até encontrar uma que emita resultados satisfatórios.

Para entendermos melhor a necessidade da utilização de camadas extras na **rede neural artificial** vamos considerar o caso do operador lógico **XOR**. Antes da invenção do recurso matemático de **camadas ocultas** no **Perceptron** a rede simplesmente repetia o cálculo na mesma estrutura e isso impedia a assimilação de padrões **não lineares** uma vez que quando utilizamos o mesmo cálculo repetidas vezes os resultados tendem a fornecer valores muito próximos dos anteriores seguindo uma progressão geralmente **linear** quando distribuímos esses resultados em um plano cartesiano.

Observe por exemplo o caso do operador lógico **AND** onde a resposta só será “verdadeira” quando ambos os testes comparativos forem verdadeiros. Aqui representaremos o “**verdadeiro**” com **1** e o “**falso**” com **0** por que como já vimos anteriormente as **redes neurais artificiais** são representações digitais de uma estrutura analógica, por isso só aceitam números/dígitos.

AND		
0	0	0
0	1	0
1	0	0
1	1	1

Para todas as possibilidades envolvendo combinações entre **0** e **1** em duas entradas, só teremos **1 (verdadeiro)** quando as duas entradas forem iguais a **1 (verdadeiro)**. Em um reconhecimento **linear** só precisamos de uma única linha para separar os padrões de saída **0** dos padrões de saída **1**, veja:

AND		
0	0	0
0	1	0
1	0	0
1	1	1

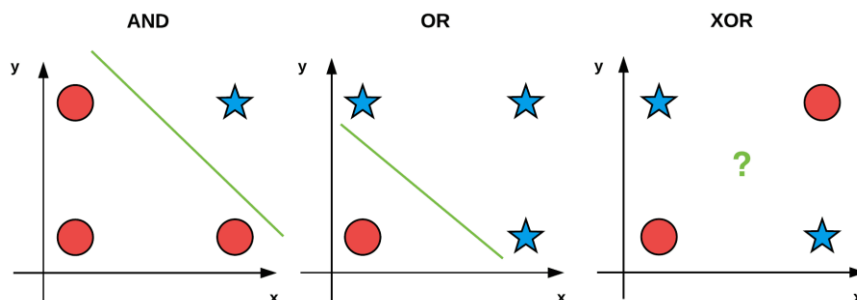
Agora vamos comparar com a distribuição do operador lógico **OR**, no operador **OR** teremos “**verdadeiro**” se pelo menos uma das alternativas for “**verdadeira**”.

OR		
0	0	0
0	1	1
1	0	1
1	1	1

Também com uma única linha é possível separar um padrão do outro, mas e se estivéssemos utilizando o operador lógico **XOR** que só retorna **verdadeiro** se apenas uma das alternativas for **verdadeira**?

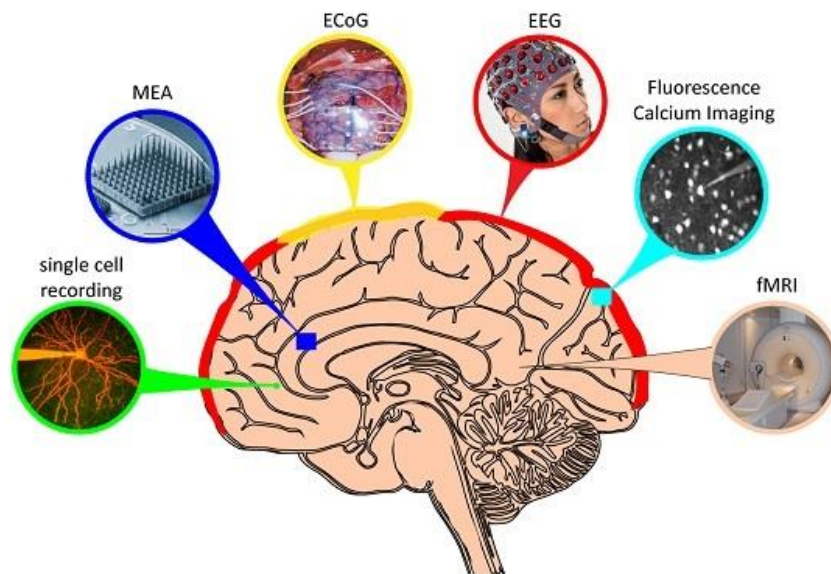
XOR		
0	0	0
0	1	1
1	0	1
1	1	0

Observe que agora foram necessárias duas linhas para separar os padrões do **0** dos padrões do **1**. Isso por que agora não temos mais dois padrões, o acima e o abaixo da linha quando a progressão era **linear** evoluindo de somente **0** para somente **1**. Dessa vez temos uma progressão **não linear**, pois os dados evoluem de somente **0** para somente **1** e retornam para somente **0** novamente, não seguindo uma ordem contínua (**linear**). Essa linha extra é conseguida matematicamente adicionando uma camada a mais na rede que retira a linearidade do cálculo ao inserir **pesos** diferentes dos da camada de entrada, com uma camada extra o algoritmo é obrigado a multiplicar o produto das entradas (**inputs**) pelos **pesos (w)** por outros **pesos** que não estão preocupados em respeitar o valor dessa multiplicação esperando uma soma que resulte na saída. Isso adicionará um aspecto de **não linearidade** nos resultados que poderá ser aplicado ao reconhecimento de padrões. Veja como ficaria essa comparação em um plano cartesiano:



Na imagem acima estamos representando o **1** com a estrela azul e o **0** com o círculo vermelho.

Monitoramento da Atividade Cerebral



* **Ressonância Magnética Funcional ou Functional Magnetic Resonance Imaging (fMRI)**: é a técnica mais utilizada na área da saúde para esse tipo de monitoramento e monitora a atividade cerebral com base no fluxo sanguíneo e na concentração de sangue em regiões específicas do cérebro. O risco de rejeição é zero por que a técnica não é invasiva e não é necessário cirurgia, porém exige um equipamento grande e pesado que só funciona com a imobilidade do paciente, o monitoramento gera uma imagem que representa exclusivamente o momento específico da captura, geralmente utilizado para detectar lesões.

* **Eletroencefalografia, Eletroencefalograma ou Electroencephalography (EEG)**: é a segunda técnica mais utilizada na saúde e a primeira em pesquisas, ela monitora a atividade elétrica do cérebro em grupos de neurônios através de eletrodos colocados por cima do couro cabeludo. O risco de rejeição é zero por que assim como o fMRI também não necessita de cirurgia, o equipamento é compacto, de fácil manuseio e portabilidade, é capaz de capturar registros por períodos maiores do que o fMRI porém a sua precisão nos resultados é inferior.

* **Eletrocorticografia ou Electrocorticography (ECoG)**: é semelhante ao EEG, também monitora a atividade elétrica em grupos de neurônios porém de forma invasiva com eletrodos diretamente conectados ao cérebro a fim de se obter resultados mais precisos e por isso só é possível mediante cirurgia contendo um relativo risco de rejeição pelo organismo.

* **Imagem de Fluorescência de Cálcio ou Fluorescence Calcium Imaging**: essa técnica mede a atividade neural através das concentrações de cálcio nas células. Diferente das técnicas que medem a atividade elétrica se concentrando nos picos de tensão momentânea como o EEG, a imagem do cálcio intracelular consegue monitorar a atividade de alterações mais lentas emitindo um panorama geral do comportamento cerebral em períodos mais longos. O

processo é invasivo, exige cirurgia e possui um considerável risco de rejeição sendo mais comumente utilizado no estudo de cobaias animais.

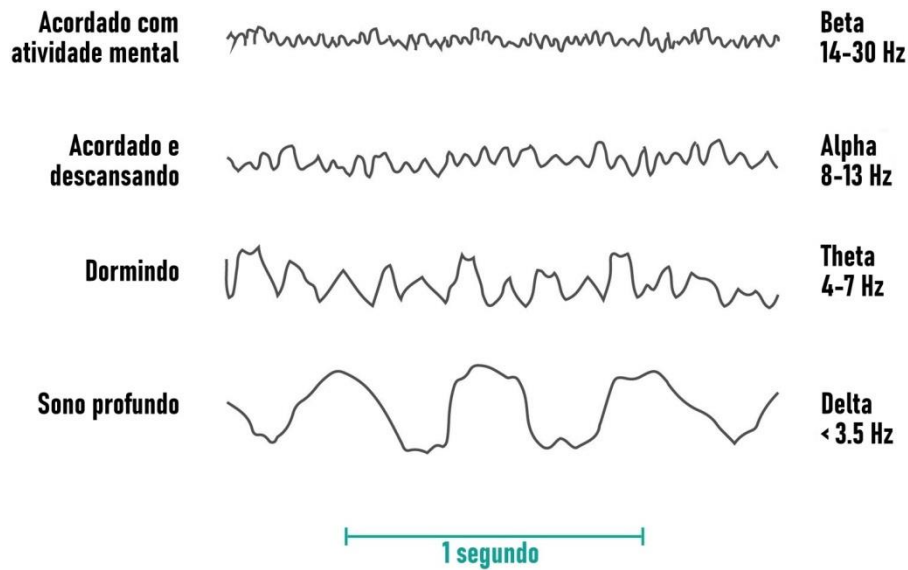
* **Matriz de Microeletrodos ou Microelectrode Array (MEA)**: essa técnica usa a implantação cirúrgica de chips com microeletrodos no cérebro para realizar uma comunicação com ou sem fio entre o cérebro e um hardware monitorando a atividade elétrica de grupos menores de neurônios, o que a torna uma técnica mais precisa do que as demais, porém com um alto risco de rejeição fisiológica sendo possível somente mediante cirurgia altamente especializada.

* **Gravação unicelular ou Single Cell Recording**: é uma técnica de monitoramento cerebral que implanta eletrodos de forma invasiva, possível somente através de cirurgia mirando a captação de sinal de um único neurônio de interesse. É usada em pesquisas com animais para estudar o funcionamento particular dos neurônios e sua anatomia. Possui um alto risco de rejeição fisiológica, mas o seu nível de precisão é muito superior ao das demais técnicas.

Na neurociência computacional dentre todas as técnicas mencionadas a mais utilizada é o eletroencefalograma (EEG), por ser não invasivo e possuir resultados relativamente satisfatórios. A maioria dos sinais elétricos do cérebro operam **entre 1 e 20 Hertz** que corresponde a faixa de sinal onde o EEG consegue performar melhor. Os sinais elétricos podem ser interpretados como ondas e com o EEG medimos as variáveis descritivas que representam uma onda eletromagnética como **frequência, comprimento, amplitude e período**. Podemos aplicar algoritmos de aprendizado de máquina para identificar padrões nessas variáveis e usar os padrões de determinadas atividade para acionar, mover ou executar periféricos através da mente usando aparelhos com ou sem fio. Também podemos reconhecer esses padrões somente com base nos tipos de ondas (**voltagem e/ou frequência**) captadas durante uma determinada ação do indivíduo. Nesse caso monitoramos os quatro tipos de ondas eletromagnéticas emitidas pelo cérebro que são as seguintes:

Onda	Voltagem	Frequência
B (Beta)	menor que 50 microvolts	de 13/ 14 a 30 Hz \cong
A (Alpha)	50 microvolts	de 7/ 8 a 13 Hz \cong
Θ (Theta)	50 microvolts	de 3.5/ 4 a 7 Hz \cong
Δ (Delta)	de 100 a 200 microvolts	menor que 3.5 / 4 Hz \cong

Por exemplo: podemos pedir para que o indivíduo aperte determina tecla enquanto o monitoramos com o EEG, ou caso se trate de uma pessoa com limitações motoras podemos pedir para que ela simplesmente imagine a ação de apertar a tecla (neste segundo caso o resultado não seria tão preciso quanto no primeiro). Depois avaliando os valores de **frequência, comprimento, amplitude e/ou período** poderemos utilizar um algoritmo que acione a mesma tecla quando o indivíduo reproduzir um padrão com valores semelhantes ao repetir o mesmo pensamento. Normalmente basta o padrão de frequência da onda **Beta** para que já seja possível reconhecer movimentos simples. As demais ondas estão relacionadas a pensamentos e estados mentais que não envolvem ação motora, no caso a onda **Alpha está ligada a atividade mental do descanso**, a **Theta ao sono** e a **Delta ao sono profundo**.



A maioria dos sistemas de “interface cérebro máquina” fazem uso desse princípio para mover objetos ou até mesmo reproduzir falas com a mente. Existem inclusive estudos na área utilizando esse tipo de técnica para tentar obter algum tipo de comunicação de pacientes em coma (quando as variáveis do consciente se tornam inoperáveis). É nesse ponto que entram em ação os algoritmos de reconhecimento de padrões que vimos anteriormente onde dentre eles as **redes neurais artificiais** são as que costumam apresentar os melhores resultados em tempo de execução já que elas reproduzem o mesmo padrão lógico que os neurônios que emitem os sinais que estamos tentando captar. Normalmente quando estamos tentando reconhecer padrões que já foram executados e estão armazenados em uma base de dados tabular os algoritmos de **aprendizado de máquina** baseados na matemática da mente fornecem um nível de assertividade maior, porém quando precisamos reconhecer esses padrões em tempo de execução as **redes neurais artificiais** são muito superiores no reconhecimento.

Um dos periféricos mais utilizados por desenvolvedores no monitoramento de sinais elétricos é o Emotiv (www.emotiv.com), um hardware portátil com uma API de conexão e código simples até mesmo para programadores iniciantes. Com ele é possível desenvolver aplicações com foco na manipulação de rotinas usando apenas a mente através de um framework Python de fácil programação e com repositório de código aberto no GitHub.



Armazenamento e Recuperação de Memórias

Sabemos que o cérebro faz o uso tanto de recursos digitais quanto analógicos durante o processo de computação das variáveis, sendo estes majoritariamente analógicos, podemos definir o cérebro como uma máquina analógica com funções digitais.

Mas como ele armazena e recupera as variáveis memorizadas? Como apenas uma pequena parte da informação é memorizada eletricamente, o cérebro utiliza a morfologia dos neurônios para representar a informação armazenada. Se todas as memórias fossem armazenadas eletricamente a carga elétrica contida no nosso cérebro seria tão grande que o faria queimar em uma espécie de combustão espontânea biológica da mesma forma que acontece quando despejamos uma intensa descarga elétrica em um organismo vivo, ele queima. Obviamente não somos queimados vivos quando pensamos e o motivo disso é que a maior parte da informação massiva contida no cérebro é armazenada analogicamente e o cérebro gera pequenas descargas elétricas somente quando necessário para acionar exclusivamente os neurônios que possuem a informação que precisa ser acessada e processada.

Um único neurônio gera um sinal elétrico que inicia uma reação em cadeia nos neurônios do entorno que contem a informação morfológica que será utilizada. Quando o sinal elétrico passa pelo neurônio que contém a informação é como se esse sinal copiasse o aspecto físico desse neurônio com a eletricidade e o interpretasse digitalmente, mas só no momento em que a recuperação da informação for necessária. Da mesma forma no processo de geração da informação as descargas elétricas são enviadas para células nervosas ao redor que criam “cicatrices” geradas por essas descargas se transformando em células neuronais onde o resultado dessas “cicatrices” do “choque” recebido são os dendritos e os terminais de saída. Depois que a morfologia celular é alterada para representar a informação que foi recebida, ela só receberá novas descargas (mais sutis do que a primeira que gerou as “cicatrices” para gravar a informação) quando essa morfologia precisar ser interpretada para ler os dados que foram armazenados.

Partindo desse pressuposto podemos concluir que quando pensamos nós na verdade não estamos nos lembrando mas sim criando uma nova informação totalmente do zero a partir da interpretação da morfologia dos neurônios envolvidos. É por isso que não é raro nos lembrarmos de falsas memórias ou de informações incompletas. Tudo o que somos, vivemos, pensamos, sentimos e sonhamos, todas as nossas experiências, traumas, erros e acertos estão armazenados na morfologia do aspecto físico dessas células alteradas por “cicatrices” de descargas elétricas. Como essas cicatrizes nunca serão umas iguais as outras por mais que possam ser parecidas, isso faz com que o vermelho que você viu ontem não seja o mesmo vermelho que você viu hoje e é totalmente diferente do vermelho que outra pessoa viu. Pessoas que ouvem e veem as mesmas coisas, que passam pelos mesmos eventos e tem as mesmas experiências poderão ter percepções e interpretações completamente diferentes e é isso o que nos torna únicos. Se não houvessem pequenos erros de flutuação randômica na transmissão de elétrons que causassem discrepâncias de uma transmissão para a outra nós seríamos todos iguais mentalmente. Parece contraditório, mas nossos erros são os únicos responsáveis por fazerem de nós seres únicos e insubstituíveis, são os nossos erros que nos tornam humanos.