



Artigo

Modelos para aprendizado de máquina

Explore as ideias por trás dos modelos de aprendizado de máquina e alguns dos principais algoritmos usados para cada um



Por M. Tim Jones

05 Dezembro 2017

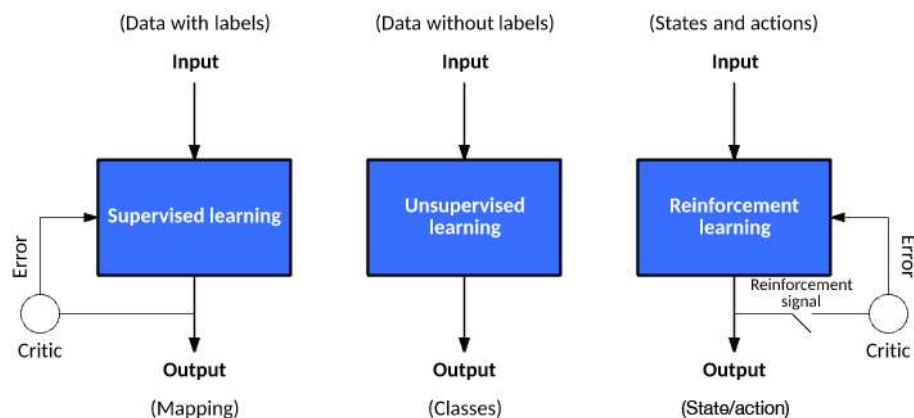
Tempo de leitura: 14 minutos

Os algoritmos usados no [aprendizado de máquina](#) se enquadram aproximadamente em três categorias: [supervisionado](#), [não supervisionado](#) e aprendizado por reforço. O *aprendizado supervisionado* envolve feedback para indicar quando uma previsão está certa ou errada, enquanto o *aprendizado não supervisionado* não envolve resposta: o algoritmo simplesmente tenta categorizar os dados com base em sua estrutura oculta. O *aprendizado por reforço* é semelhante ao aprendizado supervisionado na medida em que recebe feedback, mas não é necessariamente para cada entrada ou estado. Este tutorial explora as ideias por trás desses modelos de aprendizagem e alguns dos principais algoritmos usados para cada um.

Os algoritmos de aprendizado de máquina continuam a crescer e evoluir. Na maioria dos casos, no entanto, as preferências de cookies

tendem a se estabelecer em um dos três modelos de aprendizagem. Os modelos existem para se ajustar automaticamente de alguma forma para melhorar seu funcionamento ou comportamento.

Gráfico 1. Três modelos de aprendizagem para algoritmos



No aprendizado supervisionado, um conjunto de dados inclui suas saídas desejadas (ou *rótulos*) de tal forma que uma função pode calcular um erro para uma determinada previsão. A supervisão vem quando uma previsão é feita e um erro produzido (real vs. desejado) para alterar a função e aprender o mapeamento.

Na aprendizagem não supervisionada, um conjunto de dados não inclui uma saída desejada; portanto, não há como fiscalizar a função. Em vez disso, a função tenta segmentar o conjunto de dados em "classes" para que cada classe contenha uma parte do conjunto de dados com recursos comuns.

Finalmente, na aprendizagem por reforço, o algoritmo tenta aprender ações para um determinado conjunto de estados que levam a um estado objetivo. Um erro é fornecido não após cada exemplo (como é o caso da aprendizagem supervisionada), mas sim ao receber um sinal de reforço (como atingir o estado objetivo). Esse comportamento é semelhante ao aprendizado humano, onde o feedback não é necessariamente fornecido para todas as ações, mas quando uma recompensa é garantida.

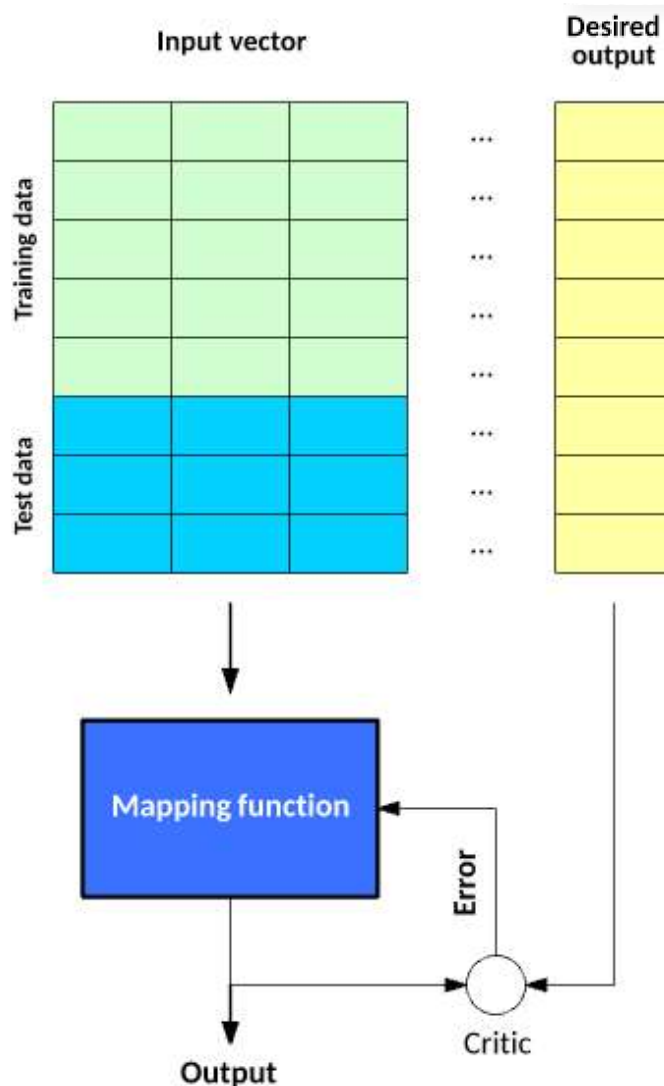
Agora, vamos nos aprofundar em cada modelo e examinar suas abordagens e algoritmos-chave.

Aprendizagem supervisionada

A aprendizagem supervisionada é o mais simples dos modelos de aprendizagem para entender. O aprendizado no modelo supervisionado envolve a criação de uma função que pode ser treinada usando um conjunto de dados de treinamento e, em seguida, aplicada a dados invisíveis para atender a algum desempenho preditivo. O objetivo é construir a função para que ela generalize bem sobre dados que nunca viu.

Você cria e testa uma função de mapeamento com aprendizado supervisionado em duas fases (veja a imagem abaixo). Na primeira fase, você segmenta um conjunto de dados em dois tipos de amostras: dados de treinamento e dados de teste. Os dados de treinamento e teste contêm um vetor de teste (as entradas) e um ou mais valores de saída desejados conhecidos. Você treina a função de mapeamento com o conjunto de dados de treinamento até que ele atinja algum nível de desempenho (uma métrica para a precisão com que a função de mapeamento mapeia os dados de treinamento para a saída desejada associada). No contexto do aprendizado supervisionado, isso ocorre com cada amostra de treinamento, onde você usa o erro (saída real versus desejada) para alterar a função de mapeamento. Na próxima fase, você testa a função de mapeamento treinada em relação aos dados de teste. Os dados de teste representam dados que não foram usados para treinamento e fornecem uma boa medida de quão bem a função de mapeamento se generaliza para dados não vistos.

Gráfico 2. As duas fases de construção e teste de uma função de mapeamento com aprendizagem supervisionada

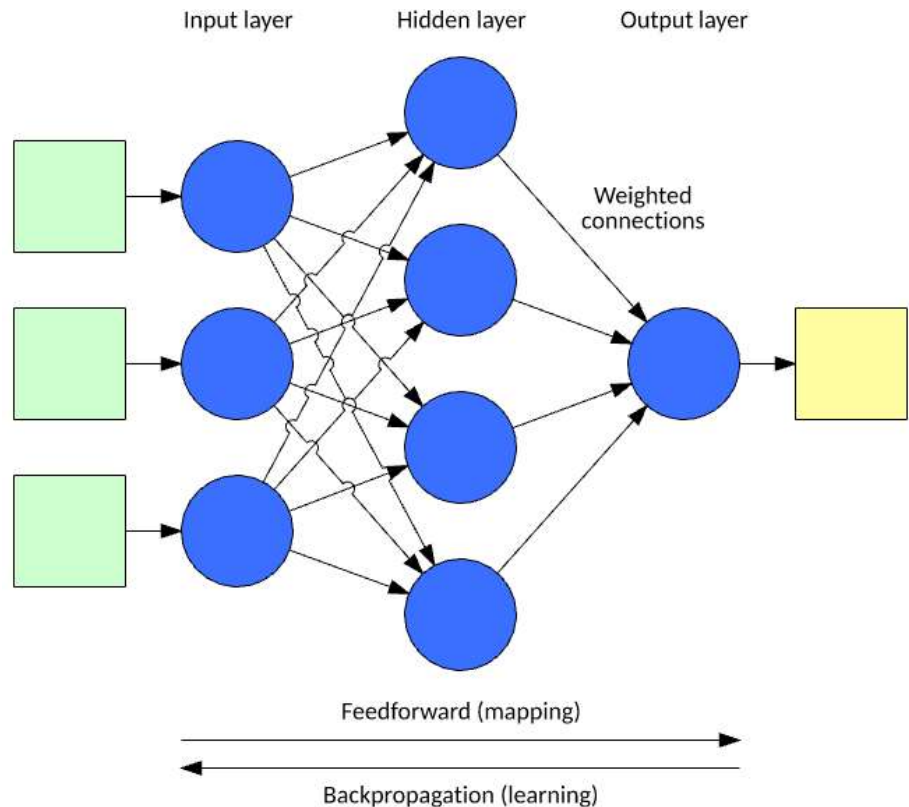


Inúmeros algoritmos existem sob o guarda-chuva do aprendizado supervisionado, como máquinas de vetor de suporte e Bayes ingênuos. Vejamos duas abordagens principais: redes neurais e árvores de decisão.

Redes neurais

Uma rede neural processa um vetor de entrada para um vetor de saída resultante através de um modelo inspirado nos neurônios e sua conectividade no cérebro. O modelo consiste em camadas de neurônios interconectados por meio de pesos que alteram a importância de certas entradas sobre outras. Cada neurônio inclui uma função de ativação que determina a saída do neurônio (em função de seu vetor de entrada multiplicado por seu vetor de peso). A saída é calculada aplicando o vetor de entrada à camada de entrada da rede, em seguida, computando as saídas de cada neurônio através da rede (de forma feed-forward).

Gráfico 3. Camadas de uma rede neural típica



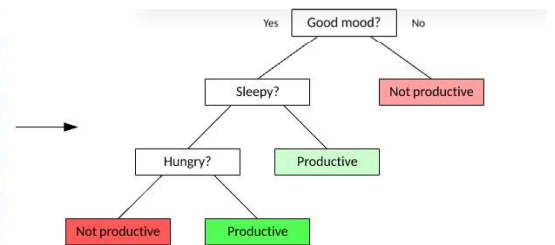
Um dos métodos de aprendizagem supervisionada mais populares para redes neurais é chamado *de retropropagação*. Na propagação de volta, você aplica um vetor de entrada e calcula o vetor de saída. O erro é computado (real vs. desejado), depois propagado de volta para ajustar os pesos e vieses a partir da camada de saída para a camada de entrada (em função de sua contribuição para a saída, ajustando para uma taxa de aprendizado). Você pode aprender mais sobre redes neurais e retropropagação em "[Um mergulho profundo em redes neurais](#)".

Árvores de decisão

Uma *árvore de decisão* é um método de aprendizagem supervisionada para classificação. Algoritmos dessa variedade criam árvores que predizem o resultado de um vetor de entrada com base em regras de decisão inferidas a partir das características presentes nos dados. As árvores de decisão são úteis porque são fáceis de visualizar para que você possa entender os fatores que levam a um resultado.

Gráfico 4. Uma árvore de decisão Preferências de cookies

Sleepy	Hungry	Good mood	Productive?
No	No	No	No
No	No	Yes	Yes
No	Yes	No	No
No	Yes	Yes	Yes
Yes	No	No	No
Yes	No	Yes	Yes
Yes	Yes	No	No
Yes	Yes	Yes	No



Existem dois tipos de modelos para árvores de decisão: *árvores de classificação*, onde a variável de destino é um valor discreto e as folhas representam rótulos de classe (como mostrado na árvore de exemplo), e *árvores de regressão*, onde a variável de destino pode assumir valores contínuos. Use um conjunto de dados para treinar a árvore, que cria um modelo a partir dos dados. Em seguida, você pode usar a árvore para tomada de decisões com dados invisíveis (descendo a árvore com base no vetor de teste atual até que uma folha seja encontrada).

Existem inúmeros algoritmos para o aprendizado da árvore de decisão. Um dos primeiros foi o Dicotomiser Iterativo 3 (ID3), que funciona dividindo o conjunto de dados em dois conjuntos de dados separados com base em um único campo no vetor. Você seleciona esse campo calculando sua entropia (uma medida da distribuição dos valores para esse campo). O objetivo é selecionar um campo do vetor que resultará em uma diminuição na entropia em divisões subsequentes do conjunto de dados à medida que a árvore é construída.

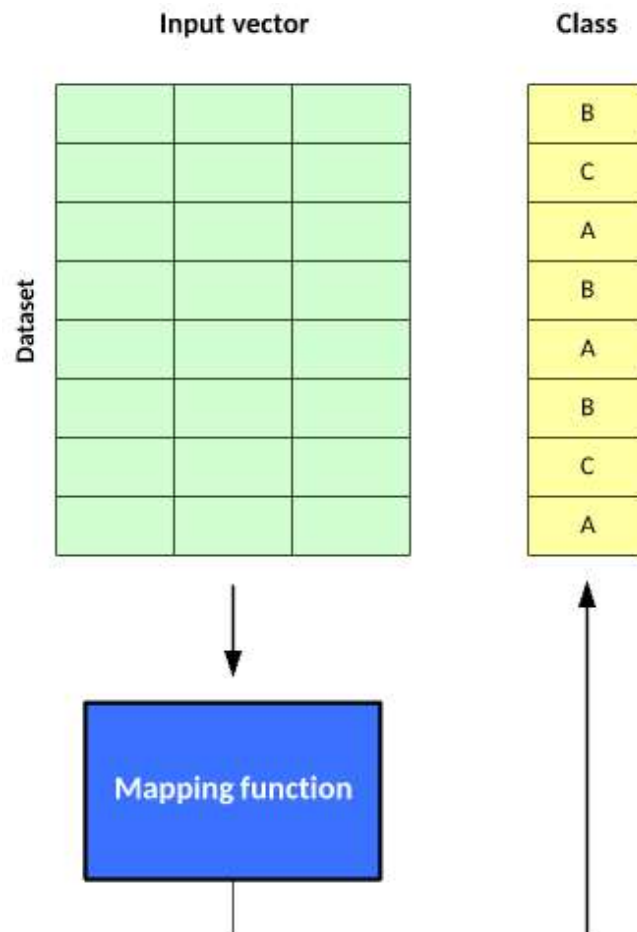
Além do ID3 está um algoritmo aprimorado chamado C4.5 (um sucessor do ID3) e Multivariate Adaptive Regression Splines (MARS), que constrói árvores de decisão com melhor manipulação de dados numéricos.

Aprendizagem não supervisionada

A aprendizagem não supervisionada também é um modelo de aprendizagem relativamente simples, mas, como o nome sugere, carece de um crítico e não tem como medir seu desempenho. O objetivo é criar uma função de mapeamento que categorize os dados em classes com base em recursos ocultos nos dados.

Assim como no aprendizado supervisionado, você usa o aprendizado não supervisionado em duas fases. Na primeira fase, a função de mapeamento segmenta um conjunto de dados em classes. Cada vetor de entrada torna-se parte de uma classe, mas o algoritmo não pode aplicar rótulos a essas classes.

Gráfico 5. As duas fases do uso da aprendizagem não supervisionada



A segmentação dos dados em classes pode ser o resultado (a partir do qual você pode tirar conclusões sobre as classes resultantes), mas você pode usar essas classes ainda mais, dependendo do aplicativo. Um desses aplicativos é um sistema de recomendação, onde o vetor de entrada pode representar as características ou compras de um usuário, e os usuários dentro de uma classe representam aqueles com interesses semelhantes que podem ser usados para marketing ou recomendações de produtos.

Você pode implementar o aprendizado não supervisionado usando uma variedade de algoritmos, como *agrupamento k-means* ou teoria de ressonância adaptati

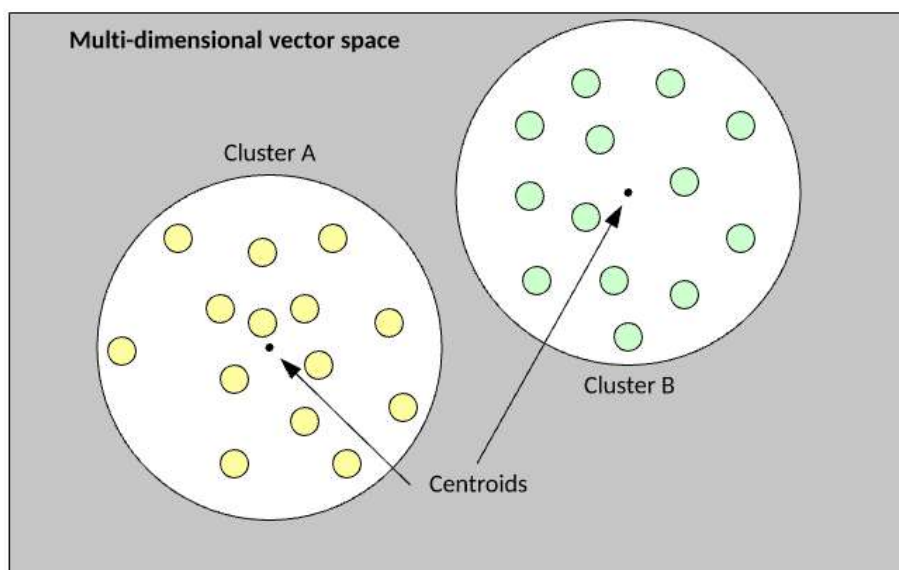
de algoritmos que implementam clustering não supervisionado de um conjunto de dados).

Agrupamento K-means

K-means clustering é um algoritmo de agrupamento simples e popular que se originou no processamento de sinais. O objetivo do algoritmo é particionar exemplos de um conjunto de dados em k clusters. Cada exemplo é um vetor numérico que permite que a distância entre vetores seja calculada como uma distância euclidiana.

O exemplo simples abaixo visualiza o particionamento de dados em $k = 2$ clusters, onde a distância euclidiana entre os exemplos é menor para o centroide (centro) do cluster, o que indica sua pertença.

Gráfico 6. Exemplo simples de agrupamento k-means



O algoritmo *k-means* é extremamente simples de entender e implementar. Você começa atribuindo aleatoriamente cada exemplo do conjunto de dados em um cluster, calcula o centroide dos clusters como a média de todos os exemplos de membro e, em seguida, itera o conjunto de dados para determinar se um exemplo está mais próximo do cluster membro ou do cluster alternativo (dado que $k = 2$). Se o membro estiver mais próximo do cluster alternati

movido para o novo cluster e seu centroide será recalculado. Esse processo continua até que nenhum exemplo se mova para o cluster alternativo.

Como ilustrado, *k-means* particiona o conjunto de dados de exemplo em clusters k sem qualquer compreensão dos recursos dentro dos vetores de exemplo (ou seja, sem supervisão).

Teoria da ressonância adaptativa

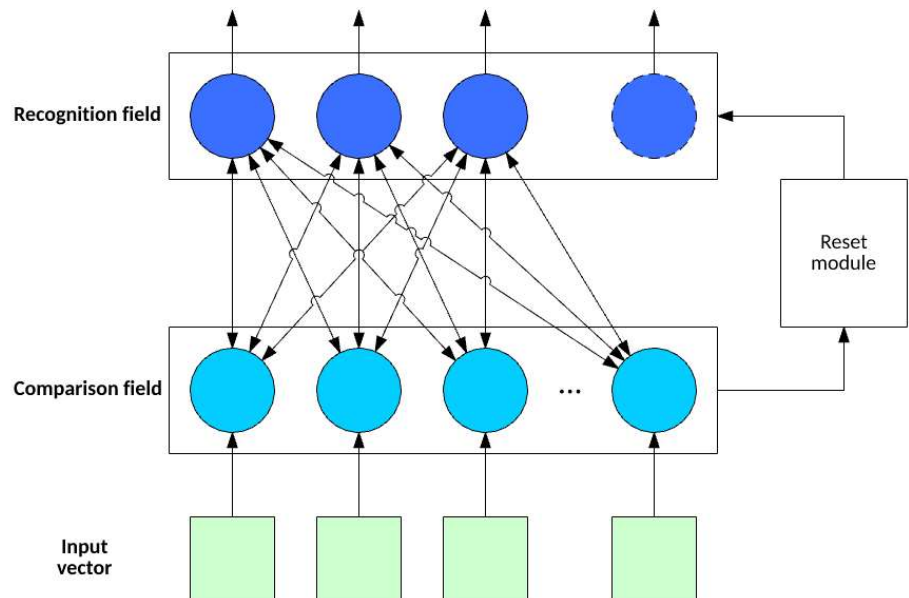
A teoria da ressonância adaptativa (TRA) é uma família de algoritmos que fornecem reconhecimento de padrões e capacidades de previsão. Você pode dividir a TRA ao longo de modelos não supervisionados e supervisionados, mas eu foco aqui no lado não supervisionado. A ART é uma arquitetura de rede neural auto-organizável. A abordagem permite aprender novos mapeamentos, mantendo o conhecimento existente.

Como *k-means*, você pode usar ART1 para clustering, mas ele tem uma vantagem fundamental em que, em vez de definir k em tempo de execução, ART1 pode alterar o número de clusters com base nos dados.

ART1 inclui três recursos principais: um campo de comparação (usado para determinar como um novo vetor de recurso se encaixa nas categorias existentes), um campo de reconhecimento (contém neurônios que representam os clusters ativos) e um módulo de redefinição. Quando o vetor de entrada é aplicado, o campo de comparação identifica o cluster no qual ele se encaixa mais. Se o vetor de entrada corresponder no campo de reconhecimento acima de um parâmetro de vigilância, as conexões com o neurônio no campo de reconhecimento serão atualizadas para dar conta desse novo vetor. Caso contrário, um novo neurônio é criado no campo de reconhecimento para dar conta de um novo cluster. Quando um novo neurônio é criado, os pesos dos neurônios existentes não são atualizados, permitindo que eles retenham o conhecimento existente. Todos os exemplos do conjunto de dados são aplicados dessa maneira até que nenhum

exemplo altere o cluster. Neste ponto, o treinamento está completo.

Gráfico 7. Características de ART1



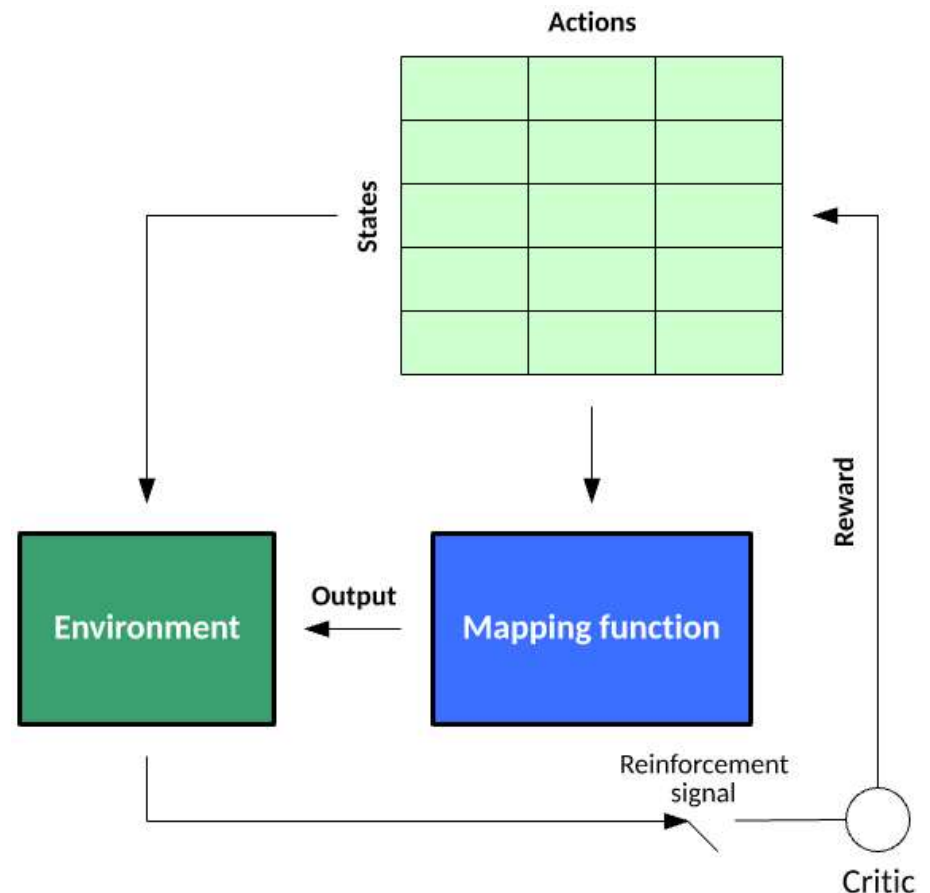
A ART inclui vários algoritmos que suportam vetores de entrada binários ou vetores de valor real (ART2). A TARV preditiva é uma variação da TARV1/2, mas depende de treinamento supervisionado.

Aprendizagem por reforço

A aprendizagem por reforço é um modelo de aprendizagem interessante, com a capacidade não apenas de aprender a mapear uma entrada para uma saída, mas de mapear uma série de entradas para saídas com dependências (processos de decisão de Markov, por exemplo). A aprendizagem por reforço existe no contexto dos estados em um ambiente e das ações possíveis em um determinado estado. Durante o processo de aprendizagem, o algoritmo explora aleatoriamente os pares estado-ação dentro de algum ambiente (para construir uma tabela de pares estado-ação), em seguida, na prática da informação aprendida, explora as recompensas do par estado-ação para escolher a melhor ação para um determinado estado que leva a algum estado objetivo. Você pode aprender mais sobre o aprendizado por reforço em ["Treinamento de uma rede neural para aprendizagem por reforço"](#)

Gráfico 8. O modelo de aprendizagem por reforço

Nesta página



Considere um agente simples que joga blackjack. Os estados representam a soma das cartas para o jogador. As ações representam o que um agente que joga blackjack pode fazer - neste caso, acertar ou ficar de pé. Treinar um agente para jogar blackjack envolveria muitas mãos de poker, onde a recompensa por um determinado nexo estado-ação é dada por ganhar ou perder. Por exemplo, o valor para um estado de 10 seria 1,0 para hit e 0,0 para stand (indicando que hit é a escolha ideal). Para o estado 20, a recompensa aprendida provavelmente seria 0,0 para acerto e 1,0 para suporte. Para uma mão menos direta, um estado de 17 pode ter valores de ação de 0,95 stand e 0,05 hit. Esse agente, então, probabilisticamente ficaria em 95% das vezes e acertaria 5% das vezes. Essas recompensas seriam apoiadas em muitas mãos de poker, indicando a melhor escolha para um determinado estado (ou mão).

Ao contrário da aprendizagem supervisionada, em que um crítico nota cada exemplo, na aprendizagem

crítico só pode fornecer uma nota quando o estado objetivo é atingido (tendo uma mão com o estado de 21).

Q-aprendizagem

Q-learning é uma abordagem para a aprendizagem por reforço que incorpora valores Q para cada par estado-ação que indicam a recompensa de seguir um determinado caminho de estado. O algoritmo geral para Q-learning é aprender recompensas em um ambiente em estágios. Cada estado engloba a tomada de ações para os estados até que um estado objetivo seja alcançado. Durante a *aprendizagem*, as ações selecionadas são realizadas de forma probabilisticamente (em função dos valores Q), o que permite a exploração do espaço de ação estatal. Quando o estado de meta é atingido, o processo começa novamente, começando a partir de alguma posição inicial.

Os valores de Q são atualizados para cada par estado-ação à medida que uma ação é selecionada para um determinado estado. O valor Q para o par estado-ação é atualizado com alguma recompensa fornecida pela movimentação (pode ser nada) junto com o valor Q máximo disponível para o novo estado alcançado pela aplicação da ação ao estado atual (descontado por um fator de desconto). Isso é ainda descontado por uma taxa de aprendizado que determina o quão valiosas as novas informações são em relação às antigas. O fator de desconto indica quão importantes são as recompensas futuras em relação às recompensas de curto prazo. Observe que o ambiente pode ser preenchido com recompensas negativas e positivas, ou apenas o estado da meta pode indicar uma recompensa.

Gráfico 9. Um algoritmo típico de Q-learning

$$Q_{st,at} = Q_{st,at} + \alpha * (r_t + \gamma * \max Q(st+1, a) - Q_{st,at})$$

The diagram shows the Q-learning update equation with labels pointing to its components:

- Learning rate**: points to α
- Reward**: points to r_t
- Discount factor**: points to γ
- New value**: points to the first $Q_{st,at}$ on the left
- Current value**: points to the $Q_{st,at}$ in the middle
- Future value estimate**: points to $\max Q(st+1, a)$

Esse algoritmo é realizado ao longo de várias épocas de alcance do estado objetivo, permitindo que os valores de Q sejam atualizados com base na seleção probabilística de ações para estados. Quando completos, os valores Q podem ser usados gananciosamente (use a ação com o maior valor Q para um determinado estado) para explorar o conhecimento adquirido para que o estado objetivo seja atingido de forma ideal.

A aprendizagem por reforço inclui outros algoritmos que têm características diferentes. State-action-reward-state-action é semelhante ao Q-learning, exceto que a seleção de uma ação não é baseada no valor *máximo* Q , mas inclui alguma probabilidade. O aprendizado por reforço é um algoritmo ideal que pode aprender a tomar decisões em um ambiente incerto.

Indo mais longe

O aprendizado de máquina se beneficia de um conjunto diversificado de algoritmos que atendem a diferentes necessidades. Os algoritmos de aprendizagem supervisionada aprendem uma função de mapeamento para um conjunto de dados com uma classificação existente, onde algoritmos de aprendizagem não supervisionados podem categorizar um conjunto de dados não rotulado com base em alguns recursos ocultos nos dados. Finalmente, a aprendizagem por reforço pode aprender políticas para a tomada de decisão em um ambiente incerto por meio da exploração iterativa desse ambiente.

Pronto para ir mais fundo? Dê uma olhada na série [Introdução ao aprendizado de máquina](#) para entender mais. [Preferências de cookies](#)

aprendizado de máquina e obter conhecimento prático das diferentes fases e tarefas.

Lenda ⓘ

Categorias ^

- Inteligência artificial Aprendizado de Máquina
- Ciência de dados Aprendizagem profunda

Interessado em IA generativa?

[Aprenda habilidades generativas de IA →](#)



Construir
Inteligente
↓
Construir
Seguro ↑

Desenvolvedor IBM

- Sobre
- Perguntas
- Freqüentes
- Aviso de terceiros

Siga-nos

- Lançamento do Twitter
- Lançamento no LinkedIn
- Lançamento do YouTube

Explorar

- Boletins
- Padrões
- Apis
- Artigos
- Tutoriais
- Projetos de código aberto
- Vídeos
- Eventos