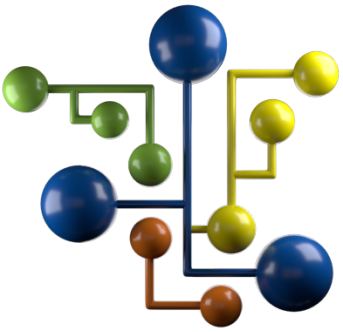


Data Science Academy

www.datascienceacademy.com.br

Deep Learning I

Regularização

Parte da mágica dos modelos de aprendizagem profunda é a regularização. Veja a definição do livro de Ian Goodfellow (Deep Learning Book): “a regularização é qualquer modificação que fazemos no algoritmo de aprendizagem, que se destina a reduzir o erro de generalização, mas não o erro de treinamento". No livro há um capítulo inteiro dedicado a este tema: http://www.deeplearningbook.org/contents/regularization.html. Mas vamos simplificar este conceito para você!

A generalização na aprendizagem de máquina refere-se a quão bem os conceitos aprendidos pelo modelo se aplicam a exemplos que não foram vistos durante o treinamento. O objetivo da maioria dos modelos de aprendizagem de máquina é generalizar bem dos dados de treinamento, a fim de fazer boas previsões no futuro para dados não vistos. O overfitting acontece quando os modelos aprendem muito bem os detalhes e o ruído dos dados de treinamento, mas não generalizam bem, então o desempenho é fraco em dados de teste. É um problema muito comum quando o conjunto de dados é muito pequeno em comparação com o número de parâmetros do modelo que precisam ser aprendidos. Este problema é particularmente crítico em redes neurais profundas onde não é incomum ter milhões de parâmetros.

A regularização é uma componente chave na prevenção do overfitting. Além disso, algumas técnicas de regularização podem ser usadas para reduzir a capacidade do modelo, mantendo a precisão, por exemplo, para direcionar alguns dos parâmetros para zero. Isso pode ser desejável para reduzir o tamanho do modelo ou diminuir o custo de avaliação em ambientes móveis onde o poder de processamento é menor.

Um dos problemas centrais em Machine Learning é como fazer um algoritmo ter bom desempenho não apenas no dataset de treino, mas também com novos dados. Muitas estratégias em Machine Learning são explicitamente desenhadas para reduzir o erro na fase de testes, permitindo uma taxa de erro maior no dataset de treino. Essas estratégias são conhecidas como regularização. Muitos profissionais não utilizam regularização, provavelmente devido a complexidade por trás do conceito. Existem muitas formas de regularização disponíveis em Deep Learning e algumas técnicas são bem recentes, como o Dropout, criado por Geofrey Hinton em 2012.

O objetivo do treinamento em redes neurais artificiais é obter uma rede que produza poucos erros no conjunto de treinamento, mas que também responda apropriadamente para novos padrões de entrada. A regularização é um método que busca melhorar a capacidade de generalização dos algoritmos de aprendizado, por meio de alguma restrição durante a fase de treinamento. A regularização ajuda a evitar o overfitting e melhora a generalização do modelo. Em Deep Learning, é muito comum o modelo aprender demais sobre as peculiaridades nos dados durante o treinamento e depois ter um baixo desempenho quando apresentado a novos dados. A regularização pode ser um remédio para este problema. Portanto:

Regularização é qualquer modificação que nós fazemos no algoritmo de aprendizagem, com o objetivo de reduzir o erro de generalização e não o erro de treinamento, a fim de evitar o overfitting.

Existem muitas estratégias de regularização. Algumas colocam restrições extras em um modelo de aprendizado de máquina, como adicionar restrições nos valores dos parâmetros. Algumas acrescentam termos extras na função objetivo que podem ser considerados como correspondentes a uma restrição suave nos valores dos parâmetros. Se escolhidas cuidadosamente, essas restrições e penalidades extras podem levar a um melhor desempenho no conjunto de dados de teste. Às vezes, essas restrições e penalidades são projetadas para codificar tipos específicos de conhecimento prévio. Outras vezes, essas restrições e penalidades são projetadas para expressar uma preferência genérica por uma classe de modelo mais simples, a fim de promover a generalização.

Na prática, um espaço de hipóteses excessivamente complexo não inclui necessariamente a função alvo ou o verdadeiro processo de geração de dados, ou mesmo uma aproximação de ambos. Nós quase nunca temos acesso ao verdadeiro processo de geração de dados, e nunca podemos saber com certeza se a família de modelos que está sendo estimada inclui o processo gerador ou não. Os algoritmos de Deep Learning são tipicamente aplicados a domínios extremamente complicados, como imagens e sequências de áudio e texto, para os quais o verdadeiro processo de geração envolve essencialmente a simulação de todo o universo de dados. O que isto significa é que controlar a complexidade do modelo não é uma questão simples de encontrar o modelo do tamanho correto, com o número correto de parâmetros. Ao invés disso, podemos encontrar - e de fato em cenários práticos de aprendizagem profunda, quase sempre achamos - que o melhor modelo (no sentido de minimizar o erro de generalização) é um grande modelo que foi regularizado.

Portanto, seu objetivo em Deep Learning é encontrar um modelo que seja grande e profundo o suficiente para representar a complexidade nos dados e que possa ser aplicado a novos conjuntos de dados, com um bom desempenho. A regularização é uma das formas usadas para se alcançar esse objetivo.

Vejamos algumas das técnicas mais comuns de regularização utilizadas atualmente em Deep Learning.

1. Dataset augmentation
2. Early stopping
3. Dropout layer
4. Regularização L1 e L2

Dataset Augmentation

Um modelo com overfitting (rede neural ou qualquer outro tipo de modelo) pode ser melhor se o algoritmo de aprendizagem processar mais dados de treinamento. Embora um conjunto de dados existente possa ser limitado, para alguns problemas de aprendizado de máquina existem maneiras relativamente fáceis de criar dados sintéticos. Para problemas de classificação geralmente é viável injetar negativos aleatórios - por ex. Imagens não relacionadas.

Não existe uma receita geral sobre como os dados sintéticos devem ser gerados e isso varia muito de um problema para outro. O princípio geral é expandir o conjunto de dados aplicando operações que refletem as variações do mundo real o mais próximo possível, melhorando o conjunto de dados, e na prática ajudando significativamente a qualidade dos modelos, independentemente da arquitetura.

Uma vez que as redes neurais profundas precisam ser treinadas em um grande número de imagens de treinamento para alcançar um desempenho satisfatório, se o conjunto de dados de imagem original contiver imagens de treinamento limitadas, é melhor aplicar Dataset Augmentation para aumentar o desempenho. Esta técnica vem sendo cada vez mais utilizada no treinamento de redes neurais profundas.

Há muitas maneiras de aplicar Dataset Augmentation, como os populares métodos de horizontally flipping, random crops e color jittering. Além disso, você pode tentar combinações de vários processos diferentes, por exemplo, fazendo a rotação e escala aleatória ao mesmo tempo. O principal objetivo deste método de regularização, é evitar o overfitting quando o conjunto de dados é muito menor do que o número de parâmetros que devem ser treinados no modelo.

Early Stopping

As “paradas mais cedo” do procedimento de treinamento evitam o overfitting, finalizando o treinamento quando o desempenho do modelo em um conjunto de validação começa a se deteriorar (como vimos ao longo deste capítulo). Um conjunto de validação é um conjunto de exemplos que nunca usamos na descida do gradiente, mas que também não faz

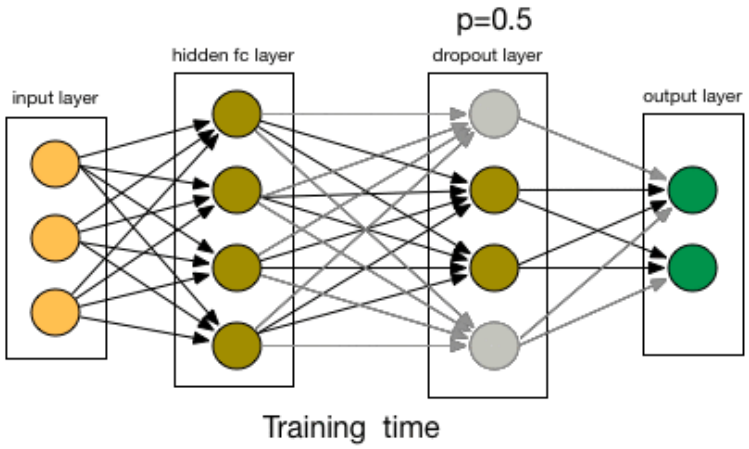
parte do conjunto de testes. Os exemplos de validação são considerados representativos de exemplos de teste futuros.

Intuitivamente, à medida que o modelo vê mais dados e aprende os padrões e correlações, tanto o treinamento como o erro de teste diminuem. Após suficientes passadas sobre os dados de treinamento, o modelo pode começar a superar e aprender o ruído no conjunto de treinamento. Neste caso, o erro de treinamento continuaria diminuindo enquanto o erro de teste (o quão bem generalizamos) ficaria pior. O Early Stopping interrompe o treinamento no momento em que o erro de validação começa a subir.

Dropout Layer

O Dropout é uma técnica de regularização criado por Geofrey Hinton em 2012, portanto bem recente, que ajuda a mudar a saída dos neurônios de uma rede neural profunda, e que pode ser aplicado em qualquer camada das redes neurais profundas. O Dropout desativa alguns dos neurônios da camada associada com alguma probabilidade p. Desativar um neurônio significa mudar o valor de saída para 0. No final os neurônios que sofreram dropout tem os parâmetros reajustados, multiplicados por p (que é a probabilidade). O efeito de usar esse algoritmo é similar ao de fazer uma média de todos os possíveis modelos da rede neural que usam um subconjunto dos parâmetros disponíveis na camada afetada.

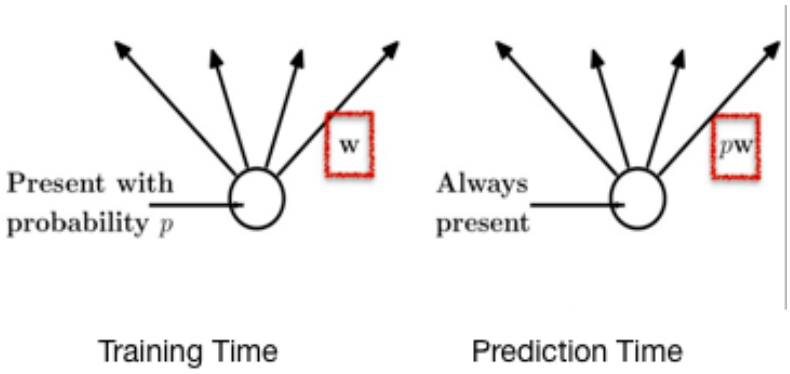
Em cada iteração de treinamento, uma camada de Dropout remove aleatoriamente alguns nós na rede, juntamente com todas as suas conexões de entrada e de saída. O Dropout (ou abandono, em uma tradução livre) pode ser aplicado a camada oculta ou de entrada.



Por que o Dropout funciona:

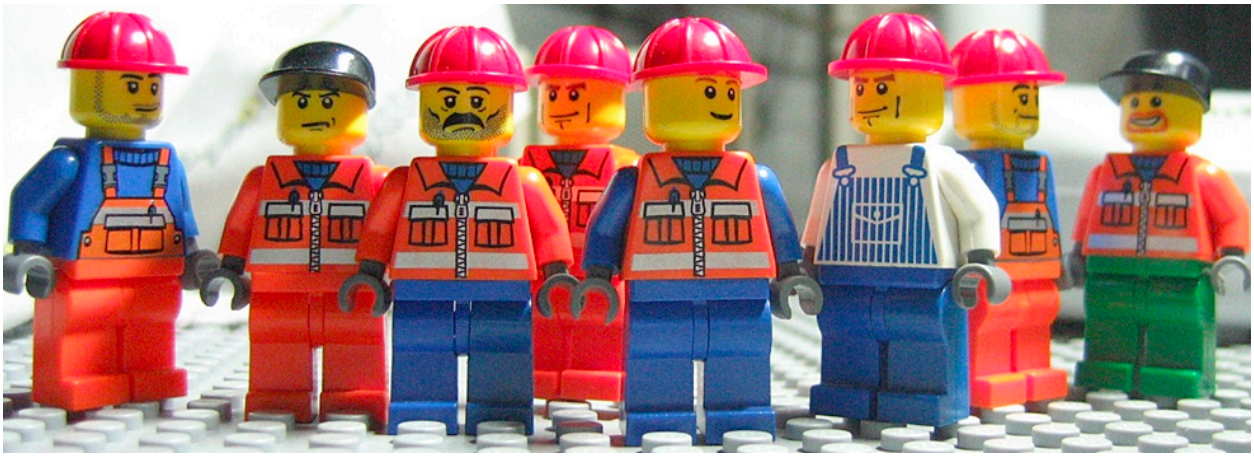
Os nós tornam-se mais insensíveis aos pesos dos outros nós (coadaptáveis) e, portanto, o modelo é mais robusto.

O Dropout pode ser visto como uma forma de modelagem média de modelos múltiplos ("ensemble"), técnica que mostra melhor desempenho na maioria das tarefas de aprendizado de máquina (treinamento ensemble é a técnica usada em modelos de RandomForest e Gradient Boosting). Se você está se perguntando como funciona a técnica ensemble, o truque é a partilha dos pesos entre todos os modelos da amostra, o que significa que cada modelo é muito fortemente regularizado pelos outros. Com este método, não precisamos treinar modelos separados que, em geral, dá muito mais trabalho, mas ainda recebemos alguns dos benefícios dos métodos ensemble.



Imagine que você tenha uma equipe de trabalhadores e o objetivo geral é aprender a construir um prédio. Quando cada um dos trabalhadores é excessivamente especializado, se alguém ficar doente ou cometer um erro, todo o projeto será gravemente afetado. A solução proposta pela técnica de "Dropout" é escolher aleatoriamente todas as semanas alguns dos

trabalhadores e enviá-los para uma viagem de negócios. A esperança é que a equipe em geral ainda aprenda como construir o edifício e, portanto, seria mais resiliente a faltas ou ao período de férias dos trabalhadores.



Devido à sua simplicidade e eficácia, o Dropout é usado hoje em várias arquiteturas, geralmente imediatamente após camadas completamente conectadas. Algumas considerações práticas de uso:

- O valor típico para p (probabilidade de manter a unidade) é > = 0.5. P torna-se outro hiperparâmetro, portanto, encontrar o valor certo depende também do problema e do conjunto de dados.

- Para as camadas de entrada, a probabilidade de manter um neurônio deve ser muito maior.

- Frameworks como Tensorflow ou Caffe2 já vêm com uma implementação de camada de Dropout.

Aqui você encontra o paper original do Dropout:

https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf

Regularização L1 e L2

A regularização L1 e L2 basicamente penalizam os coeficientes, mas ambas possuem diferentes propriedades e são usadas de diferentes maneiras. A magnitude dos coeficentes é penalizada e os erros são minimizados entre os valores previstos e os valores observados.

A penalidade de peso é uma forma padrão de regularização, amplamente utilizada na formação de outros tipos de modelo. Baseia-se fortemente na suposição implícita de que um modelo com pesos pequenos é de alguma forma mais simples do que uma rede com grandes pesos. As penalidades tentam manter os pesos pequenos ou inexistentes (zero) a menos que existam grandes gradientes para neutralizá-lo, o que torna os modelos também mais interpretáveis. Um nome alternativo na literatura para ponderações de peso é "queda de peso" (weight decay), uma vez que força os pesos a diminuir em direção a zero.

Regularização L2

Penaliza o valor quadrado do peso (o que também explica o "2" do nome). Tende a conduzir todos os pesos para valores menores.

Regularização L1

Penaliza o valor absoluto do peso. Tende a conduzir alguns pesos para exatamente zero (introduzindo sparsity no modelo), enquanto permite que alguns pesos sejam grandes. Os diagramas abaixo mostram como os valores dos pesos modificam quando aplicamos diferentes tipos de regularização.

A regularização é um tema central da aprendizagem de máquina e pode melhorar significativamente os resultados do treinamento. Não se esqueça de ajustar a regularização em seus modelos! Veremos como fazer isso em Deep Learning!