Lista 9 — Artigo

Motivação

Tenho em minhas mãos diretamente da biblioteca da UFMG o livro “Deep Learning", dos autores: Ian **GOODFELLOW**, Yoshua Bengio e Aaron Courville. Em sua seção 6.1, eles dão um exemplo de XOR através de ReLU (*rectfied linear unit*). Basta trocar nosso $g_1(x) = \tanh x$ por $g_2(x) = \max\{0, x\}$. Pouco depois, na seção 6.3.1, eles propõem 3 generalizações: *absolute value rectification*; *leaky ReLU*; PReLU (*parametric ReLU*); e finalmente o que me interessou **maxout unit**, baseado em divisão em grupos ou *dropout*, de cada grupo se tira um máximo, e ainda contém redundâncias que ajudam seus múltiplos filtros a resistir a um fenômeno chamado *catastrophic forgetting*, no qual as redes neurais se esquecem de como fazer tarefas já treinadas no passado. Isso me levou ao artigo de 9 páginas referenciado da seguinte forma:

GOODFELLOW, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013) Maxout networks. In S. Dasgupta and D. McAllester, editors, ICML'13, p. 1319-1327.

Metodologia

Trata-se de de uma arquitetura *feed-forward*, tal qual um perceptron multicamada (MLP), que usa um novo tipo de função de ativação: Dada uma entrada $x \in \mathbb{R}^d$ (x pode ser uma camada escondida; preferencialmente inicia-se com um vetor v de entrada e aplica-se o *dropout*, então $x = D(v)$. Seja a saída $y = F(v)$, sobre a qual tenta-se fazer previsões), uma camada escondida **maxout** implementa a função

$$h_i(x) = \max_{j \in [1,k]} z_{ij},$$

em que $z_{ij} = x^\top W_{\ell ij} + b_{ij}; W \in \mathbb{R}^{d \times m \times k}; b \in \mathbb{R}^{m \times k}$ são parâmetros de aprendizagem.

Uma unidade **maxout** em particular pode ser interpretada como aquela que faz uma aproximação seccionalmente linear (por exemplo o módulo de

uma variável) a uma função convexa arbitrária, por exemplo a parábola com concavidade para cima.

Consequências

A representação que produz não é esparsa, mas o *dropout* vai artificialmente tornar esparsa a representação efetiva durante o treinamento.

Redes **maxout** são um aproximador universal, de forma análoga a um MLP padrão com unidades escondidas suficientes. Cada unidade **maxout** individual pode ter uma quantidade arbitrária de componentes afins. Um modelo **maxout** com apenas 2 unidades escondidas podem aproximar arbitrariamente bem qualquer função contínua de $v \in \mathbb{R}^n$.

O autor utiliza a hipótese de domínio compacto. Teorema de aproximação de Stone-Weierstrass. É praticamente dividir uma curva em segmentos; uma superfície em planos; etc. Não parece a derivada primeira? Por isso, um caminho matemático que eu percebo é aumentar um grau, ou seja, trabalhar com formas quadráticas, elipses, parábolas, hipérboles. De Jacobianos para Hessianas.

Base de dados MNIST: Maxout MLP + dropout teve erro de teste de 0,94%; não foi nem a melhor, nem a pior performance. Foi aceitável. Entretanto, convolution + maxout + dropout teve erro de teste de 0,45% (mínima do estado da arte).

Base de dados CIFAR-10: Convolution + maxout + dropout teve erro de teste de 11,68%. Se utilizar *data augmentation* o erro cai para 9,38% (mínima do estado da arte).

Base de dados CIFAR-100: Convolution + maxout + dropout teve erro de teste de 38,57% (mínima do estado da arte).

Base de dados SVHN: Convolution + maxout + dropout teve erro de teste de 2,47% (mínima do estado da arte).

Fazer o máximo com o zero piora a acurácia. Também fica pior com tangente hiperbólica e com ReLUs. O **maxout** não tem problemas de gradiente e saturação, como os ReLUs.

Vinicius Claudino Ferraz, 31/julho/2021.