CS231n Convolutional Neural Networks for Visual Recognition

3.最適化:確率勾配降下法

序論

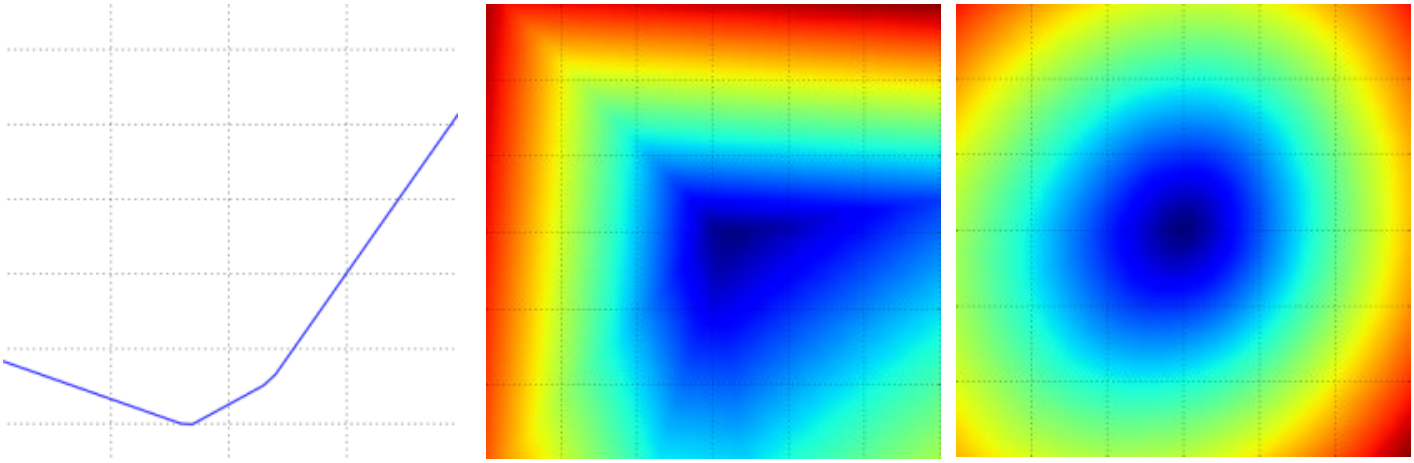
前のセクションでは、画像分類タスクの文脈で2つの重要な要素を紹介しました。
・生の画像のピクセルをクラスのスコアにマッピングする（パラメータ化された）スコア関数（例：線形関数)。
・誘導されたスコアが訓練データの正解ラベル(教師データ) とどれだけ一致しているかに基づいて、特定のパラメータセットの品質を測定する損失関数。
これには多くの方法やバージョンがあることがわかりました（例：Softmax/SVM）。具体的には、線形関数が$f(x_i, W) = Wx_i$として定式化され、開発したSVMは

$$L = \frac{1}{N}\sum_i \sum_{j \neq y_i} [\max(0, f(x_i; W)_j - f(x_i; W)y_i) + 1)] + \alpha R(W)$$

例x_iの予測値が基底真理ラベルy_iと一致するようなパラメータWの設定では、損失Lが非常に小さくなることがわかりました。ここで、3番目で最後の重要な要素を紹介します。**最適化**です。最適化とは、損失関数を最小化するパラメータWのセットを見つけるプロセスです。
ちょっと先取り：これら3つのコアコンポーネントがどのように相互作用するかを理解したら最初の構成要素（パラメータ化された関数の写像）を再検討し、線形写像よりもはるかに複雑な関数に拡張していきます。最初にニューラルネットワーク全体、次に畳み込みニューラルネットワークにです。損失関数と最適化プロセスは相対的にみて変更されません。

損失関数の可視化

この授業で見ていく損失関数は、通常、非常に高次元の空間（例えば、CIFAR-10では、線形分類器の重み行列のサイズは[10 x 3073]で、合計30,730個のパラメータ）で定義されているため、可視化するのが難しいです。しかし，線（1次元）や平面（2次元）に沿って高次元空間をスライスすることで，1つの空間についていくつかの感覚的理解を得ることができます. 例えば、ランダムな重み行列W（空間内の1点に対応）を生成し、線に沿って進み、途中で損失関数の値を記録することができます。すなわち、ランダムな方向W_1を生成し、異なるaの値について$L(W + aW_1)$を評価することにより、この方向に沿った損失を計算することができ,この処理により、aの値をx軸、損失関数の値をy軸とする単純なプロットが生成されます。また、a, bを変化させたときの損失$L(W + aW_1 + bW_2)$を評価することで、2次元でも同じ手順を行うことができます。プロットでは、a,bはx軸とy軸に対応し、損失関数の値を色の変化で表現しています。



CIFAR-10におけるMulticlass SVM(正則化なし)の1つの例(左,中)と100個の例(右)の損失関数の表面図。左: a を変化させただけの一次元の損失。 中、右: 二次元の損失スライス、青 = 低損失、赤 = 高損失。損失関数の断片的な線形構造に注目してください。複数の例の損失は平均値と結合しているので、右のボウル形状は多くの区分線形関数のボウル（中央のようなもの）の平均値です。

損失関数の断片的な線形構造は、数学的に調査することで説明できます。一つの例を挙げると

$$L_i = \sum_{j \neq y_i} [\max(0, w_j^T x_i - w_{y_i}^T x_i) + 1]$$