1E−1

絶対値表現されたエネルギー関数の最小化

金　那美[1]　　高井　昌彰[2]　　國井　利泰[1]
1 東京大学理学部
2 北海道大学工学部

1．はじめに

ニューラル・ネットワークの応用の１つに、エネルギー関数の最小化による最適化問題の解法がある。従来の方法では、エネルギー関数が常に非負になるよう、目標値との差の２乗の形でエネルギーを表現してきた。通常ネットワークのエネルギー関数はユニット状態の２次形式で与えられるため、この方法では線形の制約しか扱えないことになる。

われわれはネットワークのエネルギー関数に高次項を導入し、非線形の制約をも扱えるように拡張した[1][2]。このエネルギー関数は、従来のエネルギー関数の自然な拡張になっており、任意の次数ｎの制約（最適化問題）を、２ｎ次のエネルギー関数の最小化によって解くことができる。しかし、ｎ次の項数は$_mC_n$個（ｍ：ユニットの個数）になるため、エネルギー関数が高次になるにしたがって、必要なユニット、およびユニット間の結合数が爆発的に増大するという欠点がある。

そこで、ｎ次の制約をｎ次のエネルギー関数で直接表せるよう、エネルギー関数を目標値との差の絶対値で表現し、これを最小化する手法を考案、実現した。これにより、ネットワーク規模の増大を大幅に抑えることが可能となった。

2．絶対値表現されたエネルギー関数の最小化

次式で与えられるｎ次の制約を解くことを考える。

$$\sum_p a_p \prod_{k \in L_p} x_k = b$$

$$x_k \quad \{0,1\},\quad L=2^N,\quad N=\{1,2,3,\cdots,n\}$$

この制約の最適解を求めるためには、以下の絶対値表現されたエネルギー関数を最小化すればよい。

$$E = |F|$$

$$F = \sum a_p \prod x_k - b$$

通常の２乗誤差を考慮したエネルギー関数をもつボルツマン型のネットワークでは、あるユニットの状態を、他のユニットからの入力の総和のみを用いて決定することにより、エネルギー最小の状態に収束させることが可能である。このとき、入力の総和はユニットの状態が変わった場合のエネルギー変化量そのままを示している。絶対値を用いた場合も、大筋は同じである。仮にエネルギー関数をＦとおき、他のユニットからの総入力ｕを計算し、エネルギーがどれだけ変化するかを調べる。ただし、実際にはＦの絶対値Ｅについて考えなければならないので、入力ｕがエネルギー変化量と一致しない場合がある。例えばＦが負の場合、実際のエネルギー変化量は−ｕとなる（図１）。したがって、各ユニットの状態遷移に際し、常にエネルギー関数Ｆの符号を考慮する必要がある。また、図２（１）のように状態０のユニットへの入力ｕが正の場合、その値が現在のエネルギー値Ｅよりも大きいと、実際のエネルギー変化量は２Ｅ−ｕになる。同様に、図２（２）のように状態１のユニットへの入力ｕが負の場合、その絶対値がＥよりも大きいと、実際のエネルギー変化量は−２Ｅ−ｕになる。

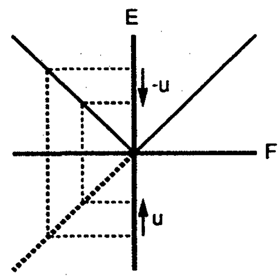


図１．Ｆが負の場合

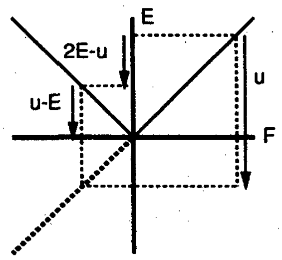


図２（１）．ユニット状態０、ｕ＞Ｅの場合

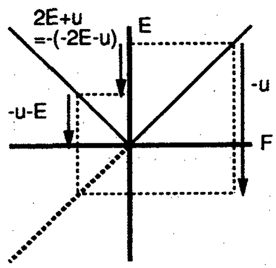


図２（２）．ユニット状態１、−ｕ＞Ｅの場合

Minimizing the absolute value energy function
Nami KIN[1], Yoshiaki TAKAI[2], Tosiyasu L. KUNII[1]
1 The University of Tokyo, 2 Hokkaido University

以上をまとめると、エネルギー関数が絶対値で与えられた場合、ボルツマン型ネットワークの逐次型のユニット状態遷移則は次のようになる：

初期設定　Fを計算する
　　$F >= 0$ ならば、
　　　sign ← 1、E ← F とする
　　$F < 0$ ならば、
　　　sign ← -1、E ← -F とする
ステップ1　ランダムにユニットx_iを選択する
ステップ2　x_iへの入力総和uを求める
$$u_i = \sum_{p \ni i} a_p \prod_{k \in p-\{i\}} x_k$$
ステップ3　sign = -1 ならば、$u_i \leftarrow -u_i$ とする
ステップ4　$x_i = 0$ かつ $u_i > E$ ならば、
　　　$u_i = 2E - u_i$、flag ← 1 とする
　　$x_i = 1$ かつ $-u_i > E$ ならば、
　　　$u_i = -2E - u_i$、flag ← 1 とする
　　上記以外の場合、flag ← 0 とする
ステップ5　次の確率Pでx_iの状態を1にする
　　　$P = 1 / (1 + \exp(-u_i/T))$
　　ただし、Tはネットワークの温度である
ステップ6　x_iの状態が変化した場合、
　　　flag = 1 ならば、
　　　　sign ← -sign とする
　　　$E \leftarrow E - u_i(2x_i - 1)$ とする

3．幾何学的制約問題への適用

われわれは、上で述べたエネルギー関数を用いて、幾何学的制約問題を解くことを試みた。[1][3]で述べたように、ニューラル・ネットワークによる解法は、データの図表化やレイアウト問題に見られるような不十分な制約に適している。幾何学的制約では、特に非線形の制約が重要な役割を果たしており、絶対値を用いたエネルギー関数の導入によって大幅にユニット間結合の数を減らすことができる。

たとえば、「線分ＡＢの長さがｄである」という２次の制約のエネルギー関数は、端点の座標をｎビットで表現することにより、次式で与えられる。

$$E = \left| \left(\sum_{i=0}^{n-1} 2^i (x_{0i} - x_{2i})\right)^2 + \left(\sum_{i=0}^{n-1} 2^i (x_{1i} - x_{3i})\right)^2 - d^2 \right|$$

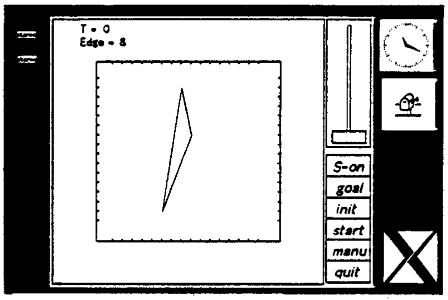
$$A:(x_a, y_a),\ B:(x_b, y_b)$$

$$x_a = \sum_{i=0} 2^i x_{0i},\ y_a = \sum_{i=0} 2^i x_{1i}$$

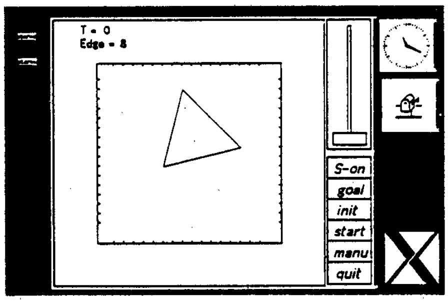
$$x_b = \sum_{i=0} 2^i x_{2i},\ y_b = \sum_{i=0} 2^i x_{3i}$$

このネットワークを用いれば、任意の線分ＡＢを初期状態として与え、状態遷移を繰り返すことで、長さｄの線分（位置は任意）に収束させることができる。更に、エネルギー関数を以下のように組み合わせれば、１辺の長さがｄの正三角形を描くことができる。

$$E = |(AB)^2 - d^2| + |(BC)^2 - d^2| + |(CA)^2 - d^2|$$



（１）初期図形



（２）収束図形

図３．プロトタイプシステムの動作例

図３は、プロトタイプシステム上で１辺が８の正三角形を描かせた例である。

4．おわりに

本論文では、従来２乗誤差の形で与えられてきたエネルギー関数を絶対値の形で表し、それによってユニット間の結合数を大幅に減らすことができることを示した。この手法は、特に非線形の制約が重要な役割を果たす幾何学的制約緩和問題において、有効である。

われわれは、Neural Networks と Computer Graphics を結び付けることにより、"Neurographics" という新しいグラフィックスの世界が開けるのではないかと期待している。今後は単なる制約緩和問題ではなく、ネットワークの特性を活かしたグラフィックス応用を行う予定である。

■参考文献

[1] N.Kin, Y.Takai, and T.L.Kunii,"A Connectionist Approach to Geometrical Constraint-Solving", (to appear in Proc. of IFIP Working Conference on Modeling in Computer Graphics, Apr. 1991).

[2] 高井、金、"幾何学的制約緩和問題へのコネクショニスト・アプローチ（１）"、情報処理学会第４１回全国大会論文集、pp.2-115～2-116、1990.

[3] 金、高井、國井、"幾何学的制約緩和問題へのコネクショニスト・アプローチ（２）"、情報処理学会第４１回全国大会論文集、pp.2-117～2-118、1990.