輪読@DL Hacks

# Difference Target Propagation

Dong-Hyun Lee, Saizheng Zhang, Asja Fischer, Antoine Biard, &
Yoshua Bengio

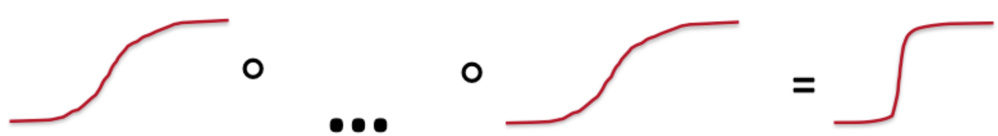
発表者　鈴木雅大

# 本論文について

- 発表学会
  - ICLR2015 workshop
- Bengio先生の最新研究？
  - 誤差逆伝播って脳ではやってないから，もっと生物学的妥当性の高い方法でやりましょうという論文．
  - Target propagationとそれを改良したdifference target propagationを提案．
  - 初出はBengio先生のテクニカルレポート(arXiv:1407.7906)．
- 本当は"Toward Biologically Plausible Deep Learning"を読みたかった．
  - 論文内のtarget propagationがわからず，こっちを読むことに．
  - 最後に軽く紹介します．
- 天才の発想に絶望．

# 誤差逆伝播について

- 最近のディープラーニングは信用割当(credit assignment)問題を解決するために誤差逆伝播(back-propagation)を利用している．
  - 信用割当問題：異なる層の誤差を明示的に知ることができない問題．パーセプトロンの学習での大きな問題だったが，誤差逆伝播で解決．
- しかしより多層になったりより強い非線形になると，より強い非線形になる．
  - 勾配が消えるor爆発的に大きくなる．
- すごい非線形になると，離散関数のようになってしまう．
  - 勾配が平らなところは0，変化するところは無限大．



# 誤差逆伝播の生物学的妥当性

誤差逆伝播はつぎのような理由で生物学的妥当性がないと考えられる.

1. 誤差逆伝播はあくまで線形であるが，生物学的には線形と非線形.
2. 脳のフィードバックが誤差逆伝播ならば，非線形なfprop計算の導関数を正確に知る必要がある.
3. bpropはfpropと対称な重みであるべき.
4. 実際のニューロンの伝達はバイナリ値（スパイク）.
5. fpropとbpropが変動するための正確な計測が必要.
6. 出力の目標値がどこから来るのか？

→ Target propagation

# 本論文の記法について

- 訓練事例の母集団の分布を $p(\mathbf{x}, \mathbf{y})$ とし，ネットワークの構造を次のように記述．

$$\mathbf{h}_i = f_i(\mathbf{h}_{i-1}) = s_i(W_i \mathbf{h}_{i-1}), \quad i = 1, \ldots, M$$

  - $h_i$:隠れ層，$h_{i-1}$:1つ下の隠れ層，$W_i$:パラメータ，$s_i$:非線形な活性化関数

- このとき，i層からj層までのパラメータを $\theta_W^{i,j} = \{W_k, k = i+1, \ldots, j\}$ とすると，$h_j$を$h_i$についての関数として記述できる．

$$\mathbf{h}_j = \mathbf{h}_j(\mathbf{h}_i; \theta_W^{i,j})$$

- 訓練事例が $(\mathbf{x}, \mathbf{y})$ のとき $L(\mathbf{h}_M(\mathbf{x}; \theta_W^{0,M}), \mathbf{y})$ を全体の誤差関数とする．

- i層に着目すると，全体の誤差関数は次のように記述できる．

$$L(\mathbf{h}_M(\mathbf{x}; \theta_W^{0,M}), \mathbf{y}) = L(\mathbf{h}_M(\mathbf{h}_i(\mathbf{x}; \theta_W^{0,i}); \theta_W^{i,M}), \mathbf{y})$$

# Targetの導入

- 誤差逆伝播ではこの誤差を各層のパラメータで偏微分して求めた誤差信号を下の層に伝えていき，全体の期待誤差を小さくする.
  - しかし深い層になると強い非線形になり，誤差が小さくなったり爆発したりする.

- この問題を解決するために各層の $\mathbf{h}_i(\mathbf{x};\theta_W^{0,i})$ を全体の誤差が小さくなるような $\hat{\mathbf{h}}_i$ に近づけることを考える．すなわち $\hat{\mathbf{h}}_i$ は次を満たす.

$$L(\mathbf{h}_M(\hat{\mathbf{h}}_i;\theta_W^{i,M}),\mathbf{y}) < L(\mathbf{h}_M(\mathbf{h}_i(\mathbf{x};\theta_W^{0,i});\theta_W^{i,M}),\mathbf{y})$$

このような $\hat{\mathbf{h}}_i$ をi層のターゲット(target)と呼ぶ.

# パラメータの更新

- ターゲット$\hat{\mathbf{h}}_i$が与えられたとき，$\mathbf{h}_i$を$\hat{\mathbf{h}}_i$に近づけることを考える．
  - これによって全体の誤差関数を小さくする．
- パラメータ $W_i$ を更新するためにlayer-localなターゲット誤差 $L_i$ を考える．

$$L_i(\hat{\mathbf{h}}_i, \mathbf{h}_i) = ||\hat{\mathbf{h}}_i - \mathbf{h}_i(\mathbf{x}; \theta_W^{0,i})||_2^2$$

- するとSGDによってパラメータは次のように更新される．

$$W_i^{(t+1)} = W_i^{(t)} - \eta_{f_i} \frac{\partial L_i(\hat{\mathbf{h}}_i, \mathbf{h}_i)}{\partial W_i}$$

$$= W_i^{(t)} - \eta_{f_i} \boxed{\frac{\partial L_i(\hat{\mathbf{h}}_i, \mathbf{h}_i)}{\partial \mathbf{h}_i}} \boxed{\frac{\partial \mathbf{h}_i(\mathbf{x}; \theta_W^{0,i})}{\partial W_i}}$$

layer-specificな学習率

$\hat{\mathbf{h}}_i$は一定

i層のパラメータを偏微分するだけ→複数の層ではないので誤差逆伝播のように強い非線形にならない

# Targetの導出

- ではどのようにして，ターゲットを求めるのか？
  - ターゲットは全体の誤差が小さくなるような値でなければならない．
  - 教師あり学習の場合，一番上の層のターゲットは明らかに全体の誤差関数の勾配から求められる．

$$\hat{\mathbf{h}}_M = \mathbf{h}_M - \hat{\eta}\frac{\partial L(\mathbf{h}_M, \mathbf{y})}{\partial \mathbf{h}_M}$$

    - $\hat{\eta}$が0.5で誤差関数がMSEならば，ターゲットはyと等しくなる．
  - 中間層では？
- [Bengio 2014]ではapproximate inverseを利用している．

# Approximate Inverse

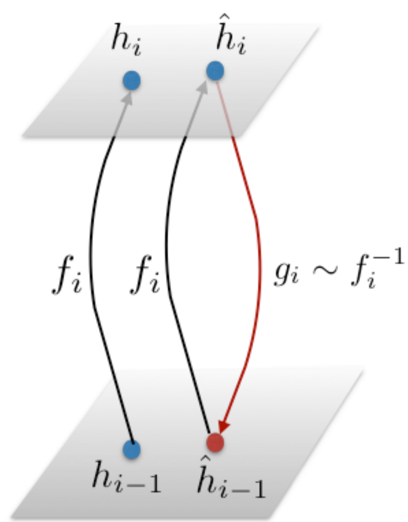
- それぞれの層の$f_i$について次のような$g_i$を考える.

$$f_i(g_i(\mathbf{h}_i)) \approx \mathbf{h}_i \quad \text{or} \quad g_i(f_i(\mathbf{h}_{i-1})) \approx \mathbf{h}_{i-1}$$

- この$g_i$を使って，i-1層のターゲットをi層のターゲットから次のように設定する.

$$\hat{\mathbf{h}}_{i-1} = g_i(\hat{\mathbf{h}}_i)$$

  - $\hat{\mathbf{h}}_{i-1}$ と $\mathbf{h}_{i-1}$ の距離を小さくしてもi番目の誤差関数が小さくなるようにする.

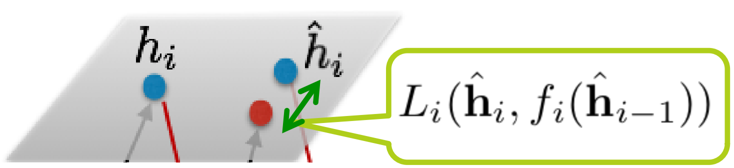


# Approximate Inverse

- もし，giが完全にfiの逆関数になるならば $L_i(\hat{\mathbf{h}}_i, f_i(\hat{\mathbf{h}}_{i-1}))$ が0になる.
  - しかし，実際に完璧な逆関数を求めることは難しい.
- よって，giをapproximate inverseとして学習（オートエンコーダーのデコーダーような感じ）.

$$g_i(\mathbf{h}_i) = \bar{s}_i(\mathbf{V}_i\mathbf{h}_i), \quad i = 1, ..., M$$

  - 各層の誤差 $L_i^{inv} = ||g_i(f_i(\mathbf{h}_{i-1})) - \mathbf{h}_{i-1}||_2^2$ を最小化してgiを得る.
- このようにすることで, $f_i(\hat{\mathbf{h}}_{i-1}) = f_i(g_i(\hat{\mathbf{h}}_i))$ は$\hat{\mathbf{h}}_i$に近くなり, $L_i(\hat{\mathbf{h}}_i, f_i(\hat{\mathbf{h}}_{i-1}))$ も小さくなる.




- また，ノイズを入れることで汎化性能を高める.

$$L_i^{inv} = ||g_i(f_i(\mathbf{h}_{i-1} + \epsilon)) - (\mathbf{h}_{i-1} + \epsilon)||_2^2, \quad \epsilon \sim N(0, \sigma)$$

# 証明1

- もし$g_i$が$f_i$の逆関数で，$f_i$が $\mathbf{h}_i = f_i(\mathbf{h}_{i-1}) = W_i s_i(\mathbf{h}_{i-1})$ となっているならば，target propagationの勾配の向きが誤差逆伝播の勾配の向きから90°以内となることが論文内で証明されている．

$$0 < \frac{1 + \Delta_1(\hat{\eta})}{\frac{\lambda_{max}}{\lambda_{min}} + \Delta_2(\hat{\eta})} \leq cos(\alpha) \leq 1$$

- 詳しい証明は論文のAppendix参照．

# Difference Target Propagation

- 実際には，逆関数が不完全だとこれまでのtargetの割り当てでは最適化に問題が発生する．

- よって，次のようなtargetを提案する．

$$\hat{\mathbf{h}}_{i-1} = \mathbf{h}_{i-1} + g_i(\hat{\mathbf{h}}_i) - g_i(\mathbf{h}_i)$$

  - これでtarget propagationする手法をdifference target propagationと呼ぶ．
  - $g_i$が$f_i$の完全な逆関数ならば普通のtarget propagationと同値．

# Difference Target Propagation

なぜdifference target propagationがいいのか？

- 安定した最適化を求めるためには，$\mathbf{h}_i$ が $\hat{\mathbf{h}}_i$ に近づくよう $\mathbf{h}_{i-1}$が $\hat{\mathbf{h}}_{i-1}$ に近づく必要がある．
  - そうしないと，上の層が最適になっているのに，下の層のパラメータを更新して全体の誤差がまた大きくなってしまう．
- よって，$\mathbf{h}_i = \hat{\mathbf{h}}_i \ \Rightarrow \ \mathbf{h}_{i-1} = \hat{\mathbf{h}}_{i-1}$ という条件によって安定化する．
  - 普通のtarget propagationでもgが逆関数ならこの条件は満たす．

$$\mathbf{h}_{i-1} = f_i^{-1}(\mathbf{h}_i) = g_i(\hat{\mathbf{h}}_i) = \hat{\mathbf{h}}_{i-1}$$

  - difference target propagationでは次の関係が成立している（1つ前のスライドの式変形により明らか）のでgが完全な逆関数でなくても成り立つ．

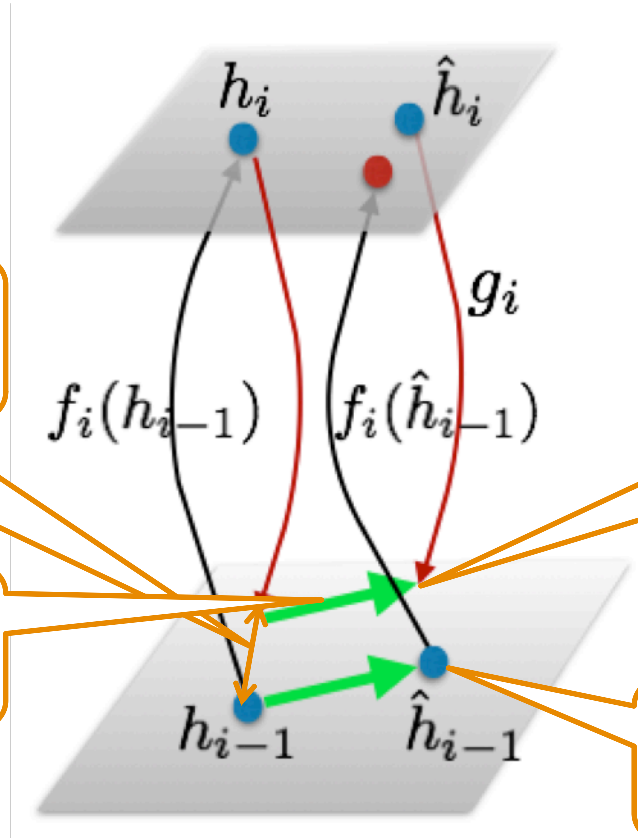
$$\hat{\mathbf{h}}_{i-1} - \mathbf{h}_{i-1} = g_i(\hat{\mathbf{h}}_i) - g_i(\mathbf{h}_i)$$

# Difference Target Propagation

直感的な理解

ここがずれてる

このベクトルを求めて




通常のtargetはここ

$h_{i-1}$に足したところをtargetとする

$$\hat{\mathbf{h}}_{i-1} = \mathbf{h}_{i-1} + g_i(\hat{\mathbf{h}}_i) - g_i(\mathbf{h}_i)$$

# 証明2

- fとgが弱い必要条件の元で, $\mathbf{h}_i$ と $\hat{\mathbf{h}}_i$ の距離が近ければ, $\mathbf{h}_{i-1}$ が $\hat{\mathbf{h}}_{i-1}$ となったとき, $\mathbf{h}_i$ も $\hat{\mathbf{h}}_i$ に近くなる．具体的には次の式が成り立つ.

$$||\hat{\mathbf{h}}_i - f_i(\hat{\mathbf{h}}_{i-1})||_2^2 < ||\hat{\mathbf{h}}_i - \mathbf{h}_i||_2^2$$

- “弱い必要条件”や証明は論文参照.

# Difference Target Propagationのアルゴリズム

**Algorithm 1** Training deep neural networks via difference target propagation

Compute unit values for all layers:
**for** $i = 1$ to $M$ **do**
  $\mathbf{h}_i \leftarrow f_i(\mathbf{h}_{i-1})$
**end for**

全ユニットを求める

Making the first target: $\hat{\mathbf{h}}_{M-1} \leftarrow \mathbf{h}_{M-1} - \hat{\eta}\frac{\partial L}{\partial \mathbf{h}_{M-1}}$, ($L$ is the global loss)
Compute targets for lower layers:
**for** $i = M-1$ to $2$ **do**
  $\hat{\mathbf{h}}_{i-1} \leftarrow \mathbf{h}_{i-1} - g_i(\mathbf{h}_i) + g_i(\hat{\mathbf{h}}_i)$
**end for**

全ターゲットを求める

Training feedback (inverse) mapping:
**for** $i = M-1$ to $2$ **do**
  Update parameters for $g_i$ using SGD with following a layer-local loss $L_i^{inv}$
  $L_i^{inv} = ||g_i(f_i(\mathbf{h}_{i-1} + \epsilon)) - (\mathbf{h}_{i-1} + \epsilon)||_2^2, \quad \epsilon \sim N(0, \sigma)$
**end for**

feedbackの訓練

Training feedforward mapping:
**for** $i = 1$ to $M$ **do**
  Update parameters for $f_i$ using SGD with following a layer-local loss $L_i$
  $L_i = ||f_i(\mathbf{h}_{i-1}) - \hat{\mathbf{h}}_i||_2^2$ if $i < M$, $L_i = L$ (the global loss) if $i = M$.
**end for**

feedforwardの訓練

# Difference Target Propagationによるオートエンコーダーの学習

- オートエンコーダーを誤差逆伝播ではなく本手法で学習する.
  - 入力層と中間層にノイズを入れたモデル.

$$\mathbf{h} = f(\mathbf{x}) = sig(W\mathbf{x} + \mathbf{b})$$

$$\mathbf{z} = g(\mathbf{h}) = sig(W^T(\mathbf{h} + \epsilon) + \mathbf{c}), \;\; \epsilon \sim N(0, \sigma)$$

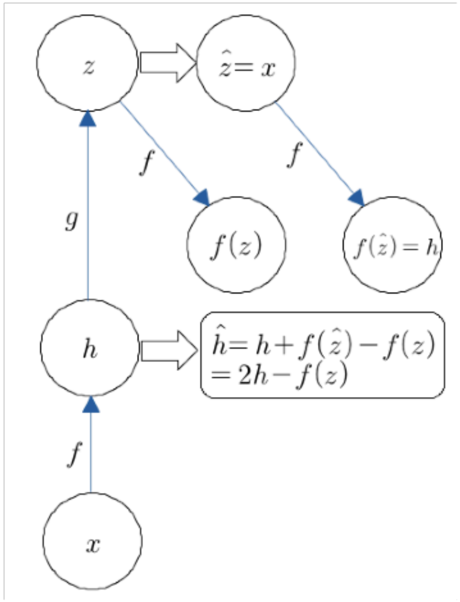
$$L = ||\mathbf{z} - \mathbf{x}||_2^2 + ||f(\mathbf{x} + \epsilon) - \mathbf{h}||_2^2, \;\; \epsilon \sim N(0, \sigma)$$

学習の流れ

1. 出力のターゲット=入力なので出力層の誤差は $L_g = ||g(\mathbf{h}) - \mathbf{x}||_2^2$ となる.
2. 隠れ層のターゲットは次のようになる（fはgのapproximate inverse）.

$$\hat{\mathbf{h}} = \mathbf{h} + f(\hat{\mathbf{z}}) - f(\mathbf{z}) = 2\mathbf{h} - f(\mathbf{z})$$

よって隠れ層の誤差は $L_f = ||f(\mathbf{x} + \epsilon) - \hat{\mathbf{h}}||_2^2$



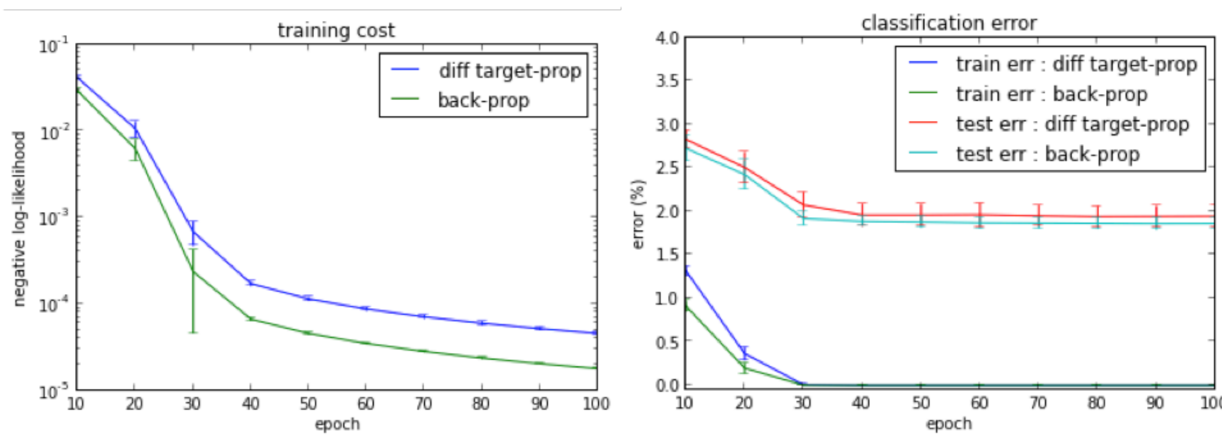
# 実験

- 次のネットワークで実験
  1. 普通の深いニューラルネットワーク
  2. ユニット間に離散的な伝達があるネットワーク
  3. 確率ニューラルネットワーク
  4. オートエンコーダー
- 初期化は重みがランダムな直交行列，バイアスは0
- 10回の平均で評価

※提案手法はdifference target propagationとする

# 1.普通の深いニューラルネットワーク

- 実験設定
  - データセット：MNIST
  - 7層で各層240ユニット
  - 活性化関数：tanh
  - その他のパラメータ等は論文参照.
- 実験結果




  - 活性化関数をReLUにするとテスト誤差がtarget:3.15%に対しback:1.62%となる.
    - ReLUは誤差逆伝播に有利であることは知られている
      (本手法には適切でない)

# 1.普通の深いニューラルネットワーク

- 実験設定
  - データセット：CIFAR-10
  - 実験設定：MNISTと同じ
  - ネットワーク構造：3072-1000-1000-1000-10
  - 入力を[0,1]に正規化，それ以外の前処理はなし
  - その他のパラメータ等は論文参照.

- 実験結果
  - テスト正解率
    - target:50.37%
    - back:53.72%
    - [Krizhevsky and Hinton 2009]は隠れ層1層+1000ユニットで49.78%，10000ユニットで51.53%
    - [Konda+ 2015] はstate-of-the-artで64.1%

# 2.ユニット間に離散的な伝達があるネットワーク

- 生物学的な考慮やニューロン間の通信コストを減らすために，次の層に伝わるときに信号を離散化する.
  - 活性化関数をステップ関数にするわけではない.

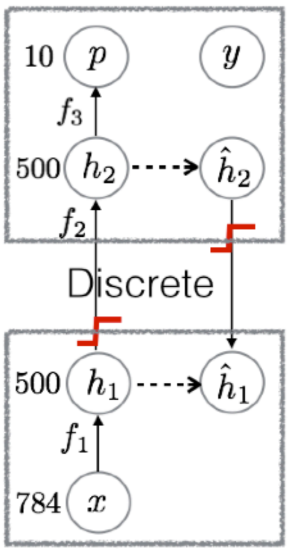
- 実験設定
  - データセット：MNIST
  - ネットワーク構造：784-500-500-10
  - 1層目から2層目で離散化

$$\mathbf{h}_2 = f_2(\mathbf{h}_1) = \tanh(W_2 sign(\mathbf{h}_1))$$

    - signは符号関数
  - 2層目の逆関数と誤差関数はつぎのようにする.

$$g_2(\mathbf{h}_2) = \tanh(V_2 sign(\mathbf{h}_2))$$

$$L_2^{inv} = ||g_2(f_2(\mathbf{h}_1 + \epsilon)) - (\mathbf{h}_1 + \epsilon)||_2^2$$

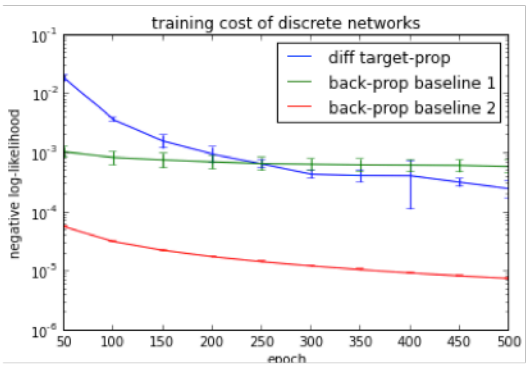


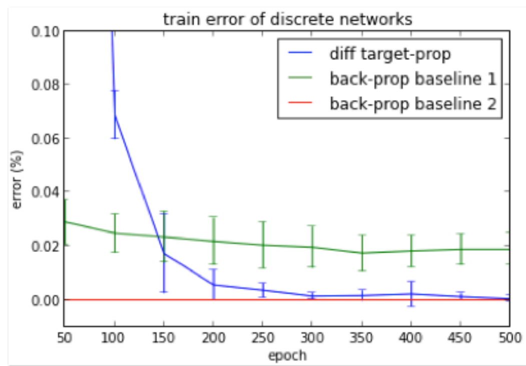
# 2.ユニット間に離散的な伝達があるネットワーク

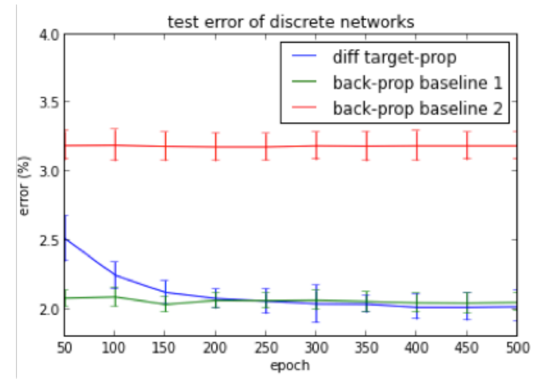
- 離散の場合，誤差逆伝播の勾配が0または微分不可能になる
- よって，次の2つのベースライン手法と比較する．
  - Straight-through estimator [Bengio+ 2013]+誤差逆伝播
    - 誤差逆伝播において，ステップ関数の導関数を無視する．
  - 離散部分より上の層を誤差逆伝播で学習（1層目は学習しない）

# 2.ユニット間に離散的な伝達があるネットワーク

- 実験結果




- Baseline1：訓練段階で0に収束しないが，汎化性能はいい.
  - 0に収束しない理由はbiased gradientで説明できる.
- Baseline2：訓練誤差やエラーは低いが，テストではよくない.
  - 1層目で意味のある表現を学習できないため.
- Target propagation：訓練段階での収束は遅いが，0に収束している. また，テストエラーも良い結果となった.
  - 離散的なネットワークでもそのまま学習できることがわかった.

# 3.確率ネットワーク

- 普通の誤差逆伝播では，確率ネットワークと離散ユニットを扱えなかった.
- 確率ネットワークは最近注目されている[Bengio 2013][Tang+ 2013][Bengio+ 2013].
  - マルチモーダルな条件付き分布P(Y | X)を学習できる.
    - 構造化された出力の予測に重要.
- 確率的なバイナリユニットは生物学的にも動機づけられる.
  - スパイキングニューロンとの類似性.

# 3.確率ネットワーク

- 実験設定：
  - データセット：MNIST
  - ネットワーク構造：784-200-200-10（[Raiko+ 2014]に従う）
  - 隠れ層は確率的バイナリユニット
  - ユニットの発火確率はシグモイド関数 $\mathbf{h}_i^p = P(\mathbf{H}_i = 1|\mathbf{h}_{i-1}) = \sigma(W_i\mathbf{h}_{i-1})$
- ベースライン手法
  - Straight-through biased gradient estimator [Bengio+ 2013]
    - 離散的なサンプリングステップの導関数は無視.

$$\delta\mathbf{h}_{i-1}^p = \delta\mathbf{h}_i^p \frac{\partial\mathbf{h}_i^p}{\partial\mathbf{h}_{i-1}^p} \approx \sigma'(W_i\mathbf{h}_{i-1})W_i^T\delta\mathbf{h}_i^p$$

# 3.確率ネットワーク

- Target propagationでは直接確率ネットワークを訓練できる.
  - ターゲットは，2層目と1層目でそれぞれ

$$\hat{\mathbf{h}}_2^p = \mathbf{h}_2^p - \eta \frac{\partial L}{\partial \mathbf{h}_2} \quad と \quad \hat{\mathbf{h}}_1^p = \mathbf{h}_1^p + g_2(\hat{\mathbf{h}}_2^p) - g_2(\mathbf{h}_2^p)$$

  - 逆関数は $g_i(\mathbf{h}_i^p) = \tanh(V_i \mathbf{h}_i^p)$で，次の誤差関数から訓練する.

$$L_i^{inv} = ||g_i(f_i(\mathbf{h}_{i-1} + \epsilon)) - (\mathbf{h}_{i-1} + \epsilon)||_2^2, \quad \epsilon \sim N(0, \sigma)$$

  - layer-localなターゲット誤差は $L_i = ||\hat{\mathbf{h}}_i^p - \mathbf{h}_i^p||_2^2$

# 3.確率ネットワーク

- 実験結果
  - 平均テストエラーでの評価

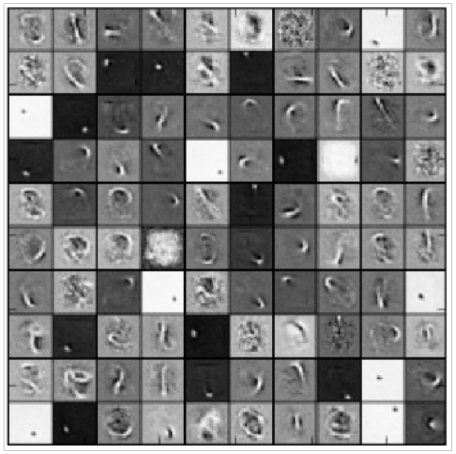
| Method | Test Error(%) |
| --- | --- |
| Difference Target-Propagation, M=1 | 1.54% |
| Straight-through gradient estimator (Bengio et al., 2013) + backprop, M=1 | |
| as reported in Raiko et al. (2014) | 1.71% |
| as reported in Tang and Salakhutdinov (2013), M=20 | 3.99% |
| as reported in Raiko et al. (2014), M=20 | 1.63% |

  - Target propagationでの結果は，MNISTによる確率ネットで最も良い結果となった.

提案手法は，確率的バイナリユニットを含んでいるネットワークでも非常に有望であることがわかった.

# 4.オートエンコーダー

- 実験設定
  - データセット：MNIST
  - ネットワーク構造：隠れ層1000ユニット
  - その他の設定は前のページで説明したとおり.
- 実験結果
  - 右のようなフィルターを学習
  - テストエラー：1.35%



誤差逆伝播による通常のオートエンコーダーと同じような結果となった.

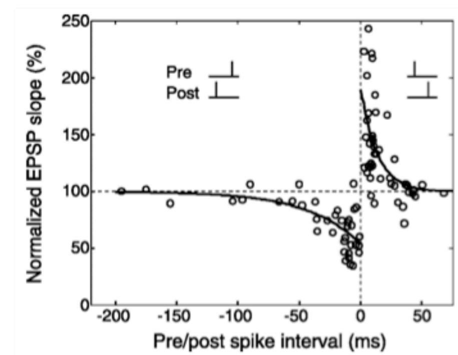
# まとめ

- 本論文では，新しい最適化手法としてtarget propagationを導入.
  - 誤差逆伝播の欠点を補い，生物学的にも妥当性がより高い.
- Difference target propagationは不完全な逆関数でもうまくいくように線形補正した手法.
- 実験では，次のことを確認した.
  - 通常の深いネットワークとオートエンコーダーで誤差逆伝播と同等の性能.
  - ユニット間に離散的な伝達があるネットワークを直接学習できる.
  - 確率ニューラルネットワークでは，MNISTでstate-of-the-art.

# Toward Biologically Plausible Deep Learning

- Hintonの2007年のtalkをベースとしている.
- スパイキングニューロンモデルで考えられている現象であるスパイクタイミング依存シナプス可塑性(STDP)に着目.
  - ニューロンの入力と出力での発火するタイミングで，重みが更新される.

$$\underbrace{\Delta W_{ij}}_{\text{synaptic change}} \propto \underbrace{S_i}_{\text{pre-spike}} \underbrace{\Delta V_j}_{\text{change in post-potential}}$$




  - この電圧の変化を目的関数Jの偏微分と見立てる.

$$\Delta V_j \approx \frac{\partial J}{\partial V_j}$$

- この論文ではJを変分EMのvariational boundとしている.

$$\log p(x) \geq E_{q^*(H|x)}[\log p(x, H)]$$

- この学習で，target propagationが使われている.

# 参考文献

- Bengio先生の公演資料
  - New York University invited talk: Towards Biologically Plausible Deep Learning
    http://www.iro.umontreal.ca/~bengioy/talks/NYU_12March2015.pdf
  - Montreal Institute of Learning Algorithms (MILA) tea-talk: Towards Biologically Plausible Deep Learning
    http://www.iro.umontreal.ca/~bengioy/talks/MILA_20Feb2015.pdf
  - NIPS'2014 MLINI workshop : Towards a biologically plausible replacement for back-prop: target-prop
    http://www.iro.umontreal.ca/~bengioy/talks/NIPS-MLINI-workshop-targetprop-13dec2014.pdf