深層ニューラルネットワークの高速化
第 5 章 蒸留 ・ 第 6 章 低ランク近似

久野 智也

2025 年 1 月 6 日

はじめに

前回までの内容

- 量子化: 浮動小数点数を整数による表現に変換する
- 低精度化: データや演算の精度を下げる
- 枝刈り: 結果に寄与しないモデルのパラメータを削除する

今回の内容

- 蒸留: 大きくて計算量の重いモデルと同じ機能をもつ小さくて計算量の軽いモデルを得る
- 低ランク近似: 行列を低ランクの行列で近似する

文献: 佐藤竜馬, "深層ニューラルネットワークの高速化", 技術評論社, 2024, p107-p158.

第 5 章 蒸留

蒸留 (distillation): 大きくて計算量の重いモデルと同じ機能をもつ小さくて計算量の軽いモデルを得るための技法

- 元となる大きなモデル: 教師モデル
- 変換先の小さなモデル: 生徒モデル

蒸留の基本的な流れ

1. 教師モデルを通常通り訓練する，または訓練済みの教師モデルをダウンロードする
2. 教師モデルの出力を模倣するようにモデルを訓練する
   -オプション: 目標タスクの教師データで生徒を訓練する

蒸留の問題設定

- 手元に教師ありデータがある場合
  - データを用いて教師モデルを通常通り訓練し，その後，知識を生徒モデルに蒸留する
  - 学習に時間がかかるが，最終的に小さくて計算量の軽いモデルさえ得られればよい場合に有用
- 手元に訓練済みの教師モデルと教師なしデータがある場合
  - 生徒モデルのみ学習すればよいので，データの準備コストや計算コストが小さい
  - 配布されている高品質な訓練済みモデルが，計算量が大きく配備できない場合に有用
- 手元に教師なしデータすらない場合
  - 生成モデルを用いて教師なしの蒸留用データを作成する

蒸留のアプローチ

蒸留には大きく分けて 2 つのアプローチがある.

- 応答蒸留 (response distillation) : 教師モデルの出力を模倣するように生徒モデルを訓練する
- 特徴蒸留 (feature distillation) : 教師モデルの中間表現を模倣するように生徒モデルを訓練する

応答蒸留 (1)

応答蒸留: 教師モデルの出力を模倣するように生徒モデルを訓練する方法
分類問題の場合には，教師モデルの出力した温度付きソフトマックス関数

$$\boldsymbol{p}^{(t,\tau=T)} \equiv \mathrm{softmax}(z^{(t)}/T) \in \mathbb{R}^{\mathcal{Y}} \tag{1}$$

をソフトラベルとし，生徒モデルも同じ温度でソフトマックス関数に掛けて

$$\boldsymbol{p}^{(s,\tau=T)} \equiv \mathrm{softmax}(z^{(s)}/T) \in \mathbb{R}^{\mathcal{Y}} \tag{2}$$

とし，これらの交差エントロピー損失

$$\ell_{\mathrm{distil}} = -T^2 \sum_{y \in \mathcal{Y}} \boldsymbol{p}_y^{(t,\tau=T)} \log \boldsymbol{p}_y^{(s,\tau=T)} \tag{3}$$

を用いて生徒モデルを訓練する.

- $z^{(t)} \in \mathbb{R}^{\mathcal{Y}}$: 教師モデル出力したベクトル
- $z^{(s)} \in \mathbb{R}^{\mathcal{Y}}$: テストモデルが出力したベクトル
- 温度 T: ハイパーパラメータ，訓練時は $T = 10$ 程度，テスト時は $T = 1$ に設定

応答蒸留 (2)

手元に教師ありデータがある場合には，教師ラベル $\boldsymbol{y} \in \mathbb{R}^{\mathcal{Y}}$ を用いて生徒モデルに対して通常の交差エントロピー損失

$$\ell_{\text{sup}} = -\sum_{y \in \mathcal{Y}} \boldsymbol{y}_y \log \boldsymbol{p}_y^{(s,\tau=1)} \tag{4}$$

を計算し，蒸留の損失 ℓ_{distil} と教師あり損失 ℓ_{sup} を合わせた

$$\ell = (1-\alpha)\ell_{\text{sup}} + \alpha\ell_{\text{distil}} \tag{5}$$

を最適化する．
訓練では，教師モデルの予測ラベルではなく，予測確率ベクトルを用いて生徒モデルを訓練することが非常に重要である．
予測誤差ベクトルには予測ラベルよりもはるかに多くの豊富な情報 (暗黙知) が込められている．

特徴蒸留

特徴蒸留: 教師モデルの中間表現を模倣するように生徒モデルを訓練する方法

- 特徴蒸留は応答蒸留よりもきめ細やかな知識の蒸留を実現できる

特徴蒸留の問題点

教師モデルの中間表現 $\boldsymbol{h}^{(t)} \in \mathbb{R}^{d_t}$ と生徒モデルの中間表現 $\boldsymbol{h}^{(s)} \in \mathbb{R}^{d_s}$ の次元が一致しない

この問題を克服するためにいくつかの手法が提案されている.

- FitNet : 補助的な射影モデルを用意し，生徒モデルの表現を射影モデルで変換した後に教師モデルの表現と比較する

特徴蒸留: FitNet

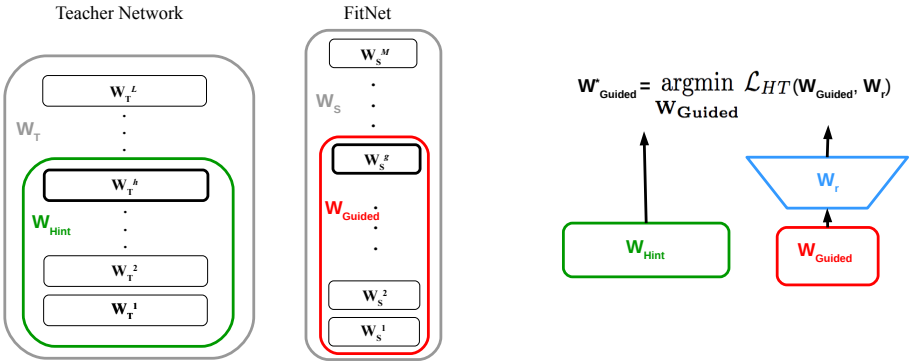

Figure: FitNet の構造 (Cited from: Adriana Romero, Nicolas Ballas, et.al, "FITNETS: HINTS FOR THIN DEEP NETS", ICLR, 2015.)

- 射影モデルを生徒モデルとともに訓練することで，生徒モデルの中間表現は，そこから教師モデルの中間表現が予測できるような，情報に富んだものになる

生徒モデルの選び方

生徒モデルは要件に合わせて選ぶことが重要である．

- GPU を用いて推論する場合：小さな行列演算も大きな行列演算も処理時間がほとんど変わらない
  → 浅くて幅が広いモデルを生徒とすることで，推論時間を大きく抑えることができる
- CPU を用いて推論する場合：並列性はあまり考慮する必要がない
  → パラメータ効率のよい深いモデルを用いることで，同じ性能であってもパラメータ数を削減でき，処理時間を短くすることができる

教師モデルと生徒モデルの容量があまりにもかけ離れている場合にはうまく蒸留ができないことがある．→ 中間的な容量の補助モデルを用意し多段階で蒸留を行う（教師アシスタント）

推論に必要なデータの変更

蒸留により推論に必要なデータを変更することで，実行速度を向上させるだけでなく，データを取得するコストを下げることができる．
例として次のようなものがある．

- 教師モデル：大きな RGB 画像を用いて画像分類をするモデル
- 生徒モデル：小さな白黒画像を入力として画像分類をするモデル
- 各 RGB 大画像 x を教師モデルに入力して教師の予測 $\boldsymbol{p}^{(t)}$ を計算し，この画像を小さくリサイズしてモノクロ化した画像 $x^{'}$ を生徒モデルに入力して生徒の予測 $\boldsymbol{p}^{(s)}$ を計算する
- 教師と生徒の予測が一致するように蒸留を行うと，小さな白黒画像のみから画像分類を行う生徒モデルを得ることができる
- 生徒モデルは低品質のデータで高速に動作する

この例は監視カメラの画像解析などで有用なアプローチとなっている．

生成モデルの利用

教師なしデータを集めることが困難な場合は，生成モデルを用いて蒸留用のデータを作成する．蒸留においては，教師モデルの入出力関係 $x \longrightarrow y$ が重要であるため，生成モデルの品質は低くてもよい．

ゼロショット知識転移 (zero shot knowledge transfer)

- 教師なしデータを集めることが困難な場合に適用できる蒸留手法
- 教師データと生徒データの出力が乖離するようなデータを最適化によって人工的に生成し，このデータでの出力が教師と一致するように生徒モデルを訓練する
- 提案論文の実験では画像分類問題において自然データを一切用いずに蒸留することに成功している

事例: FitNet

Table: FitNet による蒸留の結果

モデル	パラメータ数	乗算回数	速度向上	分類精度
教師 (5 層)	900 万 (×1)	7.3 億	×1	90.18 %
FitNet1 (11 層)	2.5 万 (×36)	0.3 億	×13.4	89.01 %
FitNet2 (11 層)	8.6 万 (×10)	1.1 億	×4.64	91.06 %
FitNet3 (13 層)	16 万 (×5.6)	3.9 億	×1.37	91.10 %
FitNet4 (19 層)	25 万 (×3.6)	3.8 億	×1.52	91.61 %

- FitNet は，訓練のしやすい浅く幅の広いモデルを訓練し，これを教師モデルとして，深く幅の狭いモデルに知識蒸留する
- 中間層の特徴を合わせる特徴蒸留を行ったのち，応答蒸留を行う

なぜ蒸留でうまくいくのか

- 大きなモデル（教師モデル）の暗黙知でガイドすることで，小さいモデル（生徒モデル）でも良い解を見つけることができる
- 訓練目標のノイズを削減することができる
  - 教師モデルの出力は決定的であるため，生徒モデルは蒸留時に矛盾に対処する必要がない
- 教師モデルのソフトラベルを使うことが正則化として有効である

蒸留のまとめ

- 蒸留によって大きいモデルと同じ機能を持つ小さいモデルを得る
  - 応答蒸留：教師モデルの出力を模倣するように生徒モデルを訓練
  - 特徴蒸留：教師モデルの中間表現を模倣するように生徒モデルを訓練
- FitNet の蒸留の結果から，速度と性能のトレードオフが優れている
- 枝刈りと同様に，生徒モデルの規模を調整することで推論時間を制御することができ，この点は低精度化に対する優位性を持つ

低ランク近似: 行列を低ランクな行列の積で近似する
例えば,

$$\boldsymbol{C} = \begin{pmatrix} 3.97 & 3.29 & -1.72 \\ 0.90 & 0.75 & -0.36 \\ 2.21 & 1.83 & -0.95 \end{pmatrix} \approx \begin{pmatrix} 1.76 \\ 0.40 \\ 0.98 \end{pmatrix} \begin{pmatrix} 2.24 & 1.87 & -0.98 \end{pmatrix} = \boldsymbol{AB} \qquad (6)$$

のように行列 $\boldsymbol{C} \in \mathbb{R}^{3\times3}$ を $\boldsymbol{A} \in \mathbb{R}^{3\times1}$ と $\boldsymbol{B} \in \mathbb{R}^{1\times3}$ の積で近似する.

低ランク近似の基本的な流れ

1. 元となるモデルを通常通り訓練する，または訓練済みの教師モデルをダウンロードする
2. モデルのパラメータ行列や中間表現行列を低ランク行列で近似する
   -オプション: 近似後のモデルをファインチューニングする

低ランク性とは (1)

行列 $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ が低ランクであるとは，n, m 以下の $r \in \mathbb{Z}_{\geq 0}$ と，$n \times r$ の行列 $\boldsymbol{A}$ と $r \times m$ の行列 $\boldsymbol{B}$ を用いて

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{B} \tag{7}$$

と表現できることである．
このように表現できる最小の $r \in \mathbb{Z}_{\geq 0}$ を $\boldsymbol{X}$ のランクと呼ぶ．
低ランク行列 $\boldsymbol{X}$ とベクトル $\boldsymbol{v} \in \mathbb{R}^m$ の積は

$$\boldsymbol{X}\boldsymbol{v} = \boldsymbol{A}(\boldsymbol{B}\boldsymbol{v}) \tag{8}$$

となる．右から順番に計算することで，$O((n+m)r)$ 時間で計算できる．
通常は $O(nm)$ 時間かかるため，r が小さい場合には大きな高速化が期待できる．

低ランク性とは (2)

現実世界に登場する多くの行列 $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ は，厳密なランクは $\min(n, m)$ であるが，低ランク行列 $\boldsymbol{X}'$ を用いて

$$\boldsymbol{X} \approx \boldsymbol{X}' = \boldsymbol{A}\boldsymbol{B} \tag{9}$$

と近似できることがしばしばある．このような近似を低ランク近似と呼び，

$$\boldsymbol{X} \approx \boldsymbol{A}\boldsymbol{B} \tag{10}$$

という表現 $\boldsymbol{A}, \boldsymbol{B}$ を得ることを行列分解と呼ぶ．
低ランク近似することは，$\boldsymbol{X}$ の列ベクトルを $\boldsymbol{A}$ の列ベクトルで張られる低次元空間に射影することに相当する．

特異値分解

特異値分解 (singular value decomposition; SVD) を用いることで，適切な近似ランクと行列分解を計算できる．

特異値分解

行列 $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ の特異値分解は

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{11}$$

となる．ここで，$\boldsymbol{U} \in \mathbb{R}^{n \times n}$, $\boldsymbol{V} \in \mathbb{R}^{m \times m}$ は直行行列，$\boldsymbol{\Sigma} \in \mathbb{R}^{n \times m}$ は対角行列である．

$\boldsymbol{\Sigma}$ の対角成分 $\sigma_1, \ldots, \sigma_{\min(n,m)}$ は非負であり，これらを特異値と呼ぶ．

特異値分解による低ランク近似

特異値 $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min(n,m)} \geq 0$ は大きいものから順に並んでいるとする．
特異値分解を r 番目の特異値で打ち切り，

$$\boldsymbol{X} \approx \boldsymbol{U}_{:,:r}\boldsymbol{\Sigma}_{:r,:r}\boldsymbol{V}_{:,:r}^T \tag{12}$$

$$= \boldsymbol{U}'\boldsymbol{V}'^T \tag{13}$$

と近似できる．

- $\boldsymbol{U}_{:,:r} \in \mathbb{R}^{n\times r}$ と $\boldsymbol{V}_{:,:r} \in \mathbb{R}^{m\times r}$: $\boldsymbol{U}$ と $\boldsymbol{V}$ の最初の r 列を取り出した行列
- $\boldsymbol{\Sigma}_{:r,:r} \in \mathbb{R}^{r\times r}$: $\boldsymbol{\Sigma}$ の最初の r 行と r 列を取り出した行列

また，$\boldsymbol{U}'$ と $\boldsymbol{V}'^T$ は次のように定義できる．

$$\boldsymbol{U}' \equiv \boldsymbol{U}_{:,:r}\boldsymbol{\Sigma}_{:r,:r}^{1/2} \in \mathbb{R}^{n\times r} \tag{14}$$

$$\boldsymbol{V}' \equiv \boldsymbol{V}_{:,:r}\boldsymbol{\Sigma}_{:r,:r}^{1/2} \in \mathbb{R}^{m\times r} \tag{15}$$

これによって，特異値を計算し，許容できる誤差までの特異値の数 r を数え，対応する $\boldsymbol{U}_{:,:r}, \boldsymbol{\Sigma}_{:r,:r}, \boldsymbol{V}_{:,:r}$ を取り出すことで低ランク近似を得ることができる．

畳み込み層

畳み込み層の出力 $\boldsymbol{Y}$ は

$$\boldsymbol{Y}_{f,h,w} = \sum_{c=1}^{C} \sum_{i=1}^{K} \sum_{j=1}^{K} \boldsymbol{X}_{c,h+i-1,w+j-1} \boldsymbol{W}_{f,c,i,j} \tag{16}$$

$$= \langle \boldsymbol{X}_{:,h:h+K,w:w+K}, \boldsymbol{W}_f \rangle \tag{17}$$

と定義される．

- 入力: $\boldsymbol{X} \in \mathbb{R}^{C \times H \times W}$
- パラメータ: $\boldsymbol{W} \in \mathbb{R}^{F \times C \times K \times K}$
- 出力: $\boldsymbol{Y} \in \mathbb{R}^{F \times (H-K+1) \times (W-K+1)}$
- K: フィルタのサイズ，C: 入力のチャンネル数，F: 出力のチャンネル数

畳み込みニューラルネットワークの低ランク近似

畳み込み層はフィルタと呼ばれる局所的な構造を抽出する機構を用いる.

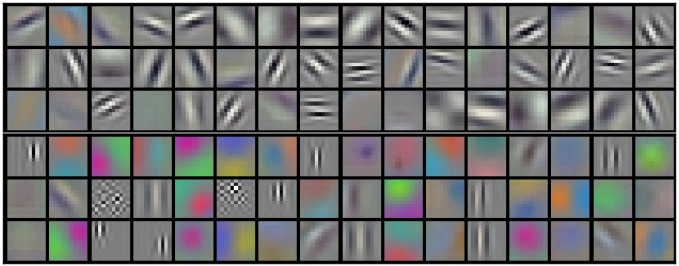

Figure: 畳み込み層フィルタの例 (Cited from: Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS, 2012.)

- ImageNet で訓練された AlexNet の 1 層目の畳み込み層のフィルタを可視化したもの
- 各パネルは 3 × 11 × 11 の一つのフィルタを表す
- 元のフィルタが 3 × 11 × 11 の自由度はフルに活用していないことが分かる

空間方向の低ランク近似

空間方向の低ランク近似:

- パラメータの空間方向 $K \times K$ 次元のパッチ $\boldsymbol{W}_{f,c,:,:} \in \mathbb{R}^{K \times K}$ を二つの行列 $\boldsymbol{A} \in \mathbb{R}^{K \times D}$, $\boldsymbol{B} \in \mathbb{R}^{D \times K}$ の積で近似する
- 行列ベクトル積の演算数およびパラメータ数は，K^2 から $2DK$ になる

フィルタ方向の低ランク近似

フィルタ方向の低ランク近似:

- フィルタの基底 $\boldsymbol{B}_1, \ldots, \boldsymbol{B}_D \in \mathbb{R}^{C \times K \times K}$ を用意し，各フィルタを

$$\boldsymbol{W}_f = \sum_{d=1}^{D} \boldsymbol{A}'_{f,d} \boldsymbol{B}_d \in \mathbb{R}^{C \times K \times K} \tag{18}$$

  というように，基底の重み付き和で表現する
  ここで，$\boldsymbol{A}' \in \mathbb{R}^{F \times D}$ は，フィルタの重みを表す行列である
- パラメータ数およびパッチあたりの演算数は，CK^2F から $(CK^2 + F)D$ になる

重み近似

- フルランクのパラメータ $\boldsymbol{W} \in \mathbb{R}^{F \times C \times K \times K}$ を低ランク近似する方法には，重み近似と出力近似の 2 種類の方針がある．

重み近似: 低ランク近似後のフィルタの重みと元のフルランクの重みの差を最小化する

- 第 1 層のパラメータを $\boldsymbol{B} \in \mathbb{R}^{D \times C \times K \times 1}$ とし，第 2 層のパラメータを $\boldsymbol{A} \in \mathbb{R}^{F \times D \times 1 \times K}$ とする
- $\boldsymbol{W}_{f,c,i,j}$ の代わりに $(\sum_{d=1}^{D} \boldsymbol{A}_{f,d,1,j} \boldsymbol{B}_{c,d,i,1})$ を用いることに対応する
- 重み近似は

$$\min_{\boldsymbol{A},\boldsymbol{B}} \sum_{i,j,c,f} \left(\boldsymbol{W}_{f,c,i,j} - \sum_{d=1}^{D} \boldsymbol{A}_{f,d,1,j} \boldsymbol{B}_{c,d,i,1} \right)^2 \tag{19}$$

  という最小化問題として定式化される

- この問題は特異値分解を用いて簡単に解くことができる

出力近似

出力近似: 低ランク近似後の出力と元のモデルの出力の差を最小化する

- 重みに低ランク性はないが，データや活性値が低ランクな場合に効果的
- 入力に近い層から一つずつ層を近似していく
- 入力を $\boldsymbol{X} \in \mathbb{R}^{(CK^2)\times N}$，当該層より以前の部分を近似したモデルに対してデータを入力したときの当該層に対する入力を $\hat{\boldsymbol{X}} \in \mathbb{R}^{(CK^2)\times N}$ とする
- 第 1 層のパラメータを $\boldsymbol{B} \in \mathbb{R}^{C\times K^2\times D}$ とし，第 2 層のパラメータを $\boldsymbol{A} \in \mathbb{R}^{F\times D}$ とする
- 近似なしの出力は $\boldsymbol{W}\boldsymbol{X}$ であり，近似ありの出力は $\boldsymbol{A}\boldsymbol{B}^T\hat{\boldsymbol{X}}$ となる
- 出力近似は

$$\min_{\boldsymbol{A},\boldsymbol{B}} \|\boldsymbol{W}\boldsymbol{X} - \boldsymbol{A}\boldsymbol{B}^T\hat{\boldsymbol{X}}\|_F^2 \tag{20}$$

  という最小化問題として定式化される
- この問題も特異値分解により解くことができる

注意機構について (1)

注意機構: 三つのベクトルの集合を受け取り，新たなベクトルの集合を出力する関数
入力する三つのベクトルの集合を次のように表す．

- クエリ (問い合わせ): $\boldsymbol{Q} \in \mathbb{R}^{n \times d}$
- キー (鍵): $\boldsymbol{K} \in \mathbb{R}^{m \times d}$
- バリュー (値): $\boldsymbol{V} \in \mathbb{R}^{m \times d'}$

出力 $\boldsymbol{Y} \in \mathbb{R}^{n \times d'}$ は以下で表される．

$$\boldsymbol{Y}_i = \sum_{j=1}^{m} \boldsymbol{A}_{ij} \boldsymbol{V}_j \in \mathbb{R}^{d'} \tag{21}$$

$$\boldsymbol{A}_{ij} = \frac{\exp\left(\boldsymbol{Q}_i^T \boldsymbol{K}_j\right)}{\sum_{j'=1}^{m} \exp\left(\boldsymbol{Q}_i^T \boldsymbol{K}_{j'}\right)} \in \mathbb{R} \tag{22}$$

- $\boldsymbol{A}_{ij}$: クエリ $\boldsymbol{Q}_i$ に対するキー $\boldsymbol{K}_j$ の重み，$\boldsymbol{A} \in \mathbb{R}^{n \times m}$: 注意行列

注意機構について (2)

注意機構の式を行列形式で書き下すと

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{V} \in \mathbb{R}^{n \times d'} \tag{23}$$

$$\boldsymbol{A} = \mathrm{softmax}(\boldsymbol{Q}\boldsymbol{K}^T) \in \mathbb{R}^{n \times m} \tag{24}$$

となる．

- 注意機構は，機械翻訳や言語モデルなどの自然言語処理や，画像認識などのコンピュータビジョンにおいて幅広く利用されている
- 計算量とメモリ消費量が非常に大きいため，これを削減する研究が盛んに行われている

6.3.2 カーネル法

カーネル法: カーネル関数と呼ばれる類似度を測る関数をもとにした手法
カーネル関数 k: データ $\boldsymbol{x}, \boldsymbol{x}'$ を入力とし，それらの類似度を表す実数値を出力する

ガウスカーネル

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2}\right) \tag{25}$$

カーネル法と注意機構

ナダラヤ・ワトソンカーネル回帰: 回帰問題に対するカーネル法の一種

- 訓練データ $(\boldsymbol{x_1}, \boldsymbol{y_1}), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)$ を用いて以下のような予測を行う

$$\hat{y}(\boldsymbol{x}') = \frac{\sum_{i=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}')\boldsymbol{y}_i}{\sum_{i=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}')} \tag{26}$$

- $\boldsymbol{x}_i$: カーネル関数に基づいた教師データ， $\hat{y}(\boldsymbol{x}')$: テストデータ $\boldsymbol{x}'$ に対する予測値
- これを行列形式で表すと以下のようになる．

$$\hat{\boldsymbol{y}} = \boldsymbol{K}\boldsymbol{y} \in \mathbb{R}^m \tag{27}$$

$$\boldsymbol{K} = \mathrm{softmax}(-d(\boldsymbol{X}, \boldsymbol{X}')^2/2) \in \mathbb{R}^{n \times m} \tag{28}$$

- $\boldsymbol{X}$: 訓練データをまとめた行列，$\boldsymbol{y}$: 目標値をまとめたベクトル，$\boldsymbol{X}'$: テストデータをまとめた行列
- d: ユークリッド距離を出力する関数

この手法は注意機構と非常に似ている．

ランダム特徴量

ランダム特徴量: カーネル法の計算量を削減するための手法の一つ
ランダム特徴量を

$$\psi(\boldsymbol{x}) = \sqrt{\frac{2}{D}} \begin{bmatrix} \cos\left(\boldsymbol{\omega}_1^T \boldsymbol{x} + b_1\right) \\ \cos\left(\boldsymbol{\omega}_2^T \boldsymbol{x} + b_2\right) \\ \vdots \\ \cos\left(\boldsymbol{\omega}_D^T \boldsymbol{x} + b_D\right) \end{bmatrix} \in \mathbb{R}^{2D} \tag{29}$$

と定義すると，ガウスカーネル

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2}\right) \tag{30}$$

は次のように近似できる．

$$k(\boldsymbol{x}, \boldsymbol{x}') \approx \psi(\boldsymbol{x})^T \psi(\boldsymbol{x}') \tag{31}$$

ランダム特徴量を用いた注意機構の近似

$$\exp\left(\boldsymbol{Q}_i^T \boldsymbol{K}_j\right) = \exp\left(\|\boldsymbol{Q}_i\|^2/2\right) \exp\left(\|\boldsymbol{K}_j\|^2/2\right) \exp\left(-\|\boldsymbol{Q}_i - \boldsymbol{K}_j\|^2/2\right) \tag{32}$$

であるため，ガウスカーネルのランダム特徴量による近似より

$$\exp\left(-\|\boldsymbol{Q}_i - \boldsymbol{K}_j\|^2/2\right) \approx \boldsymbol{\psi}(\boldsymbol{Q}_i)^T \boldsymbol{\psi}(\boldsymbol{K}_j) \tag{33}$$

であるため

$$\boldsymbol{\psi}'(\boldsymbol{x}) \equiv \exp\left(\|\boldsymbol{x}\|^2/2\right) \boldsymbol{\psi}(\boldsymbol{x}) \tag{34}$$

と定義すると，

$$\exp\left(\boldsymbol{Q}_i^T \boldsymbol{K}_j\right) \approx \boldsymbol{\psi}'(\boldsymbol{Q}_i)^T \boldsymbol{\psi}'(\boldsymbol{K}_j) \tag{35}$$

と近似できる．

低ランク近似のまとめ

- 低ランク近似によって大きい行列を小さい行列の積で近似する
  - 畳み込み層と注意機構の低ランク近似を紹介した
- 性能を少し落とすだけで，大幅な高速化を達成できる
- 低ランク近似を仮定したアーキテクチャを用いることで従来のモデルと同等の性能を達成できる