딥러닝 및 응용 중간 고사

이름: 박찬현

학번: 2016025432

1. (10점) 다음 용어들의 의미를 서술 및 정의하세요.

a. Logistic regression:

데이터가 어떤 label에 속할 확률을 0과 1사이 값으로 예측 / 이를 토대로 가능성 더 높은 label 부여

b. Cross-entropy loss (정의):

$$L(y,\hat{y}) = -(y\log\hat{y} + (1-y)\log(1-\hat{y}))$$

c. Backpropagation:

gradient를 이용해 파라미터(ex. w,b)를 업데이트 해가며 loss를 줄이는 과정.

d. Locality in CNN:

Hidden Layer node가 모든 input을 연결하지 않고, 인접한 input들하고만 연결하여 connection을 줄이는 것. (image 일부만 필터링에 사용)

parameter↓ 학습용이

e. Weight sharing in CNN:

다른 노드와의 weight 값을 공유한다.
즉 weight가 항상 같은 filter로 convolution을 진행한다.

10

2. (10점) 다음 activation 함수들의 정의를 쓰세요.

a. Sigmoid: $\sigma(z) = \frac{1}{1+e^{-z}}$

b. Tanh: $Tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$

c. ReLu: $ReLu(x) = \max(0,x)$

d. Leaky ReLu: $Leaky\ ReLU(x) = \max(0.1x, x)$

5

3. (5점) 딥러닝에서 L2 regularization을 수학적으로 표현하고, gradient descent의 update 식과 연관하여 왜 "weight decay"로 불리는지 이유를 서술하시오.

원래 J에 $\frac{\lambda}{2m}\|W\|_2^2$을 더해준다

$$dw = (\text{from back prop}) + \frac{\lambda}{m}w$$

$$w = w - \alpha\cdot dw$$

$$= \underbrace{\left(1-\alpha\cdot\frac{\lambda}{m}\right)}_{<1}w - \alpha\cdot(\text{backprop})$$

<1 이므로 w가 줄어든다.

다만, $\lambda\uparrow \rightarrow w\downarrow \rightarrow z\downarrow$

너무 큰 λ 들어오면 $w=0$이 되어 Underfit할 가능성有

$$J(W^{[1]},b^{[1]},\dots,W^{[L]},b^{[L]}) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(\hat{y}^{(i)},y^{(i)}) + \frac{\lambda}{2m}\sum_{l=1}^{L}\|W^{[l]}\|_F^2$$

Frobenius norm: $\|W^{[l]}\|_F^2 = \sum_{i=1}^{n^{[l]}}\sum_{j=1}^{n^{[l-1]}}\left(w_{ij}^{[l]}\right)^2$ $\quad W^{[l]}: (n^{[l]}, n^{[l-1]})$

Gradient descent: $dW^{[l]} = (\text{from backprop}) + \frac{\lambda}{m}W^{[l]}$ $\quad \left(\because \frac{\partial\|W^{[l]}\|_F^2}{\partial W^{[l]}} = 2W^{[l]}\right)$

$W^{[l]} := W^{[l]} - \alpha\cdot dW^{[l]}$

10

4. (10점) 트레이닝 및 테스트 단계에서의 dropout 기법을 각각 설명하고, 왜 regularization 효과가 있는지를 서술하시오.

Training시 특정 노드를 임의로 정하여 connection을 끊고 update한다.
Test시 모든 노드를 사용한다.

regularization 효과 : 한 feature에 의존하지 않고, Weight을 spread 시키기 때문.

5

5. (10점) 아래 표에서 bias와 variance가 각각 어떻게 변화하는지를 +/- 로 표시하시오.

	bias / Variance	Bias	Variance
L1 Regularization	+ −	− +	−
Deeper & larger networks	− +	−	+ −
Dropout	+ −	− +	−
Less training data	− +	+ −	+
Dataset augmentation	+ ~~+~~	− +	−

1+6+20+28
=48+7=55

6-a. (10점) 입력 영상 x와 필터 f 가 다음과 같을 때, 다음 표의 연산을 수행하시오.

1	2	3	0	0
0	4	5	6	0
0	0	7	8	9
0	0	0	0	0
0	0	0	0	10

x

1	0	2
0	5	0
3	0	4

f

4	0	3
0	5	0
2	0	1

35
26
16 20 28
7 48 55
2+0+25+32
57

4+9+20+7= 13+27=40
12+30+14+9
=65
4+12+35
=47 (51)
27+13
15 , 16+18+35
=34+35
20+40
28+27+10
55

	Without zero-padding	With zero-padding
Correlation	56 59 90 / 10 51 45 / 14 16 65	21 30 51 15 18 / 4 56 59 90 24 / 8 16 5 45 51 / 0 14 16 65 8 / 0 0 0 0 50
Convolution	40 41 65 / 15 69 60 / 21 24 65	50 0 0 0 0 / 8 65 16 14 0 / 51 45 5 10 8 / 24 90 59 56 4 / 18 15 51 30 21

33
6+0+25+8=41

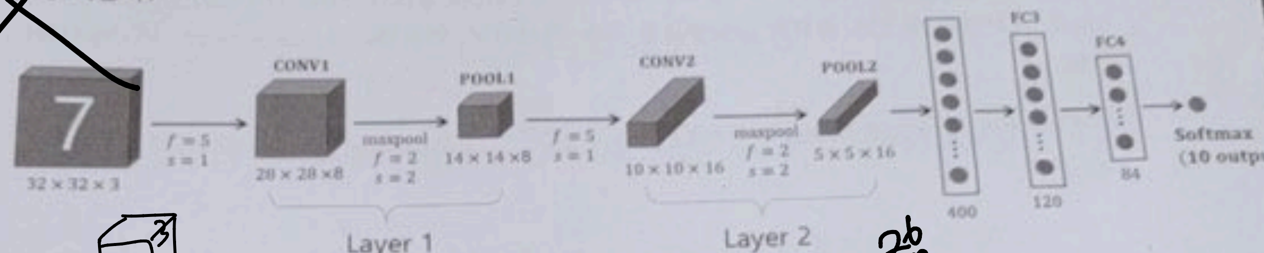
6-b. (5점) 최신 딥러닝 library에서 convolution 과 correlation을 혼동해서 사용하고 있고, 이렇게 해도 되는 이유는 무엇일까?

Correlation 결과를 ... convolution ...

똑같기 ...

CNN은 필터(F)를 찾아내는 것이 목표이므로, 계산상 간편하고 / flip-flop을 이용해 같은 F를 찾을 수 있는 correlation을 사용하는 것 convolution 계산 후엔든 것과 에 아무 문제가 없다.

7. (10점) 아래는 Lenet-5의 모델 architecture를 도식화한 것이다. 전체 학습 parameter의 수가 어떻게 되는가?



5×5
3

26
8
208

	Activation shape	#parameters
Input:	(32,32,3) 0	0 0
Conv1 (f=5, s=1)	(28,28,8) (5×5+1)×8	[redacted]
POOL1	(14,14,8) 0	0 [redacted] ×
Conv2 (f=5, s=1)	(10,10,16) (5×5+1)×16	[redacted] O
POOL2	(5,5,16) 0	0 [redacted] ×
FC3	(120,1) 400×120+1	[redacted]
FC4	(84,1) 120×84+1	[redacted] bias O.
Softmax	(10,1) 84×10+1	[redacted]

8. (10점) Gradient descent with momentum, RMSProp, ADAM optimizer를 각각 설명하시오.

Gradient descent with momentum:

$V_{dw_t} = \beta_1 dw_{t-1} + (1-\beta_1) dw_t$

$V_{db_t} = \beta_1 db_{t-1} + (1-\beta_1) db_t$

과거 몇개의 gradient의 average를 가지고 mini Batch의 경우 여러 곳으로 튀었던 gradient를 smooth하게 한다.

$V_{dw} = \beta \cdot V_{dw} + (1-\beta) dw$

// db

→ $w = w - \alpha \cdot V_{dw}$

$b = b - \alpha V_{db}$

RMSProp: 이것도 Gradient average를 ... 수직방향은 작게, 수평방향은 크게 간다

dw와 db를 $\sqrt{S_{dw}}$, $\sqrt{S_{db}}$로 나누어 update가 빠른 방향은 느리게 느린 빠르게 만든다.

$S_{dw_t} = \beta_2 dw_{t-1}^2 + (1-\beta_2) \ldots$

$S_{db_t} = \beta_2 db_{t-1}^2 + (1-\beta_2) db_t^2$

$db := \frac{db}{\sqrt{S_{db}+e^{-8}}}$

larger smaller

$S_{dw} = \beta \cdot S_{dw} + (1-\beta)(dw)^2$

$w = w - \frac{\alpha \cdot dw}{\sqrt{S_{dw}}}$

ADAM optimizer

Momentum과 RMSProp을 합치고, bias correction을 적용한다.

$$W = W - \alpha \cdot \frac{V_{dw}^{corrected}}{\sqrt{S_{dw}^{corrected}} + \varepsilon}$$

9. (10점) 입력 $(x_1, x_2, \cdots, x_n)$, 출력 o_d, loss function이 E일 때, gradient descent with momentum rule을 계산 하시오. (단, 필요한 gradient를 모두 계산하되, 이동평균의 bias correction은 무시해도 좋음)

$o_d = w_0 + w_1 x_1^{(d)} + \cdots + w_n x_n^{(d)}$ $\quad E = \sum_{d=1}^{m}(o_d - t_d)^2$

$$V_{dw} = \beta \cdot V_{dw} + (1-\beta) dw$$

$$W = W - \alpha \cdot V_{dw}$$

10. (10점) 아래와 같은 2-layered network 에서 activation function은 ReLu, loss function $E = \sum_{d=1}^{m}(o_d - t_d)^2$ 일 때 (mini batch size = m), $dW^{[1]}, dB^{[1]}, dW^{[2]}, dB^{[2]}$를 계산하고, gradient update rule을 적용하시오.

$x_1, x_2 \rightarrow \boxed{z^{[1]} = W^{[1]}x + B^{[1]}} \longrightarrow \boxed{a^{[1]} = ReLU(z^{[1]})} \rightarrow \boxed{z^{[2]} = W^{[2]}a^{[1]} + B^{[2]}} \rightarrow \boxed{a^{[2]} = ReLU(z^{[2]})}$

$\rightarrow Loss(a^{[2]}, O_d)$

$a^{[2]} = t_d$

ReLU 미분 → 1 (양수일때), 0 (음수일때)

$da^{[2]} = \frac{\partial L}{\partial a^{[2]}} = E'(a^{[2]})$

$dW^{[2]}$ $\frac{\partial L}{\partial z^{[2]}} = \frac{\partial L}{\partial a^{[2]}} \cdot \frac{\partial a^{[2]}}{\partial z^{[2]}} = E'(a^{[2]}) \cdot 1 = E'(t_d)$

$da^{[1]} = \frac{\partial L}{\partial a^{[1]}} = \frac{\partial L}{\partial z^{[2]}} \cdot \frac{dz^{[2]}}{da^{[1]}} = dW^{[2]} \cdot W$

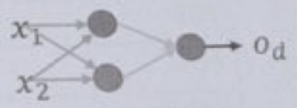
$dB^{[2]}$ $\frac{\partial L}{\partial B^{[2]}} = \frac{\partial L}{\partial z^{[2]}} \cdot \frac{\partial z^{[2]}}{\partial B^{[2]}} = dW^{[2]} \cdot 1 = dW^{[2]}$

$dz^{[1]} = \frac{\partial L}{\partial z^{[1]}} = \frac{\partial L}{\partial a^{[1]}} \cdot \frac{\partial a^{[1]}}{\partial z^{[1]}} = da^{[1]} \cdot 1 = da^{[1]} = dW^{[2]} \cdot W^{[2]}$

$dW^{[1]} = \frac{\partial L}{\partial W^{[1]}} = \frac{\partial L}{\partial z^{[1]}} \cdot \frac{\partial z^{[1]}}{\partial W^{[1]}} = dz^{[1]} \cdot X = (dW^{[2]} \cdot W^{[2]})(x_1 + x_2)$

$dB^{[1]} = \frac{\partial L}{\partial B^{[1]}} = \frac{\partial L}{\partial z^{[1]}} \cdot \frac{\partial z^{[1]}}{\partial B^{[1]}} = dz^{[1]} \cdot 1 = dz^{[1]} = dW^{[2]} \cdot W^{[2]}$

10. (10점) 아래와 같은 2-layered network 에서 activation function은 ReLu, loss function $E = \sum_{d=1}^{m}(o_d - t_d)^2$ 일 때 (mini batch size = m), $dW^{[1]}, dB^{[1]}, dW^{[2]}, dB^{[2]}$를 계산하고, gradient update rule을 적용하시오.



$$x \rightarrow \boxed{z^{[1]} = W^{[1]}x + B^{[1]}} \rightarrow \boxed{a^{[1]} = \mathrm{ReLU}(z^{[1]})} \rightarrow \boxed{z^{[2]} = W^{[2]}a^{[1]} + B^{[2]}} \rightarrow \boxed{a^{[2]} = \mathrm{ReLU}(z^{[2]})} \rightarrow \boxed{L(a^{[2]}, y)}$$

$W^{[1]}, B^{[1]}$ → $z^{[1]}$; $W^{[2]}, B^{[2]}$ → $z^{[2]}$

$$L(a^{[2]}, y) = (a^{[2]} - y)^2$$

$$\mathrm{ReLU}' = \begin{cases} 1 & (x \geq 0) \\ 0 & (\text{else}) \end{cases}$$

$$da^{[2]} = \frac{dL}{da^{[2]}} = 2(a^{[2]} - y)$$

$$dz^{[2]} = da^{[2]} \cdot \frac{da^{[2]}}{dz^{[2]}} = \begin{cases} 2(a^{[2]} - y) \cdot 1 & (z^{[2]} \geq 0) \\ 0 & \text{otherwise} \end{cases}$$

$$dw^{[2]} = dz^{[2]} \cdot \cancel{\frac{dz^{[2]}}{dw^{[2]}}}\, a^{[1]T} = \begin{cases} 2(a^{[2]} - y) \cdot a^{[1]T} & (z^{[2]} \geq 0) \\ 0 & \text{otherwise} \end{cases}$$

$$dB^{[2]} = dz^{[2]} \cdot \cancel{\frac{dz^{[2]}}{dB^{[2]}}} = \begin{cases} 2(a^{[2]} - y) & (z^{[2]} \geq 0) \\ 0 & (\text{otherwise}) \end{cases}$$

$$da^{[1]} = dz^{[2]} \cdot \frac{dz^{[2]}}{da^{[1]}} = \begin{cases} 2(a^{[2]} - y) \cdot w^{[2]} & (z^{[2]} \geq 0) \\ 0 & \text{otherwise} \end{cases}$$

$$dz^{[1]} = da^{[1]} \cdot \frac{da^{[1]}}{dz^{[1]}} = \begin{cases} 2(a^{[2]} - y) w^{[2]} & (z^{[2]} \geq 0,\ z^{[1]} \geq 0) \\ 0 & (\text{otherwise}) \end{cases}$$

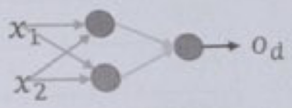
$$dw^{[1]} = dz^{[1]} \cdot \frac{dz^{[1]}}{dw^{[1]}} = \begin{cases} 2(a^{[2]} - y) w^{[2]} \cdot X & (z^{[2]} \geq 0,\ z^{[1]} \geq 0) \\ 0 & \text{otherwise} \end{cases}$$

$$dB^{[1]} = dz^{[1]} \cdot \frac{dz^{[1]}}{dB^{[1]}} = \begin{cases} 2(a^{[2]} - y) w^{[2]} & (z^{[2]} \geq 0,\ z^{[1]} \geq 0) \\ 0 & \text{otherwise} \end{cases}$$

$$W^{[1]} = W^{[1]} - \alpha \cdot dW^{[1]} \qquad W^{[2]} = W^{[2]} - \alpha \cdot dW^{[2]}$$

$$B^{[1]} = B^{[1]} - \alpha \cdot dB^{[1]} \qquad B^{[2]} = B^{[2]} - \alpha \cdot dB^{[2]}.$$

10. (10점) 아래와 같은 2-layered network 에서 activation function은 ReLu, loss function $E = \sum_{d=1}^{m}(o_d - t_d)^2$ 일 때 (mini batch size = m), $dW^{[1]}, dB^{[1]}, dW^{[2]}, dB^{[2]}$를 계산하고, gradient update rule을 적용하시오.



$$x \rightarrow \boxed{z^{[1]} = W^{[1]}x + B^{[1]}} \rightarrow \boxed{a^{[1]} = \mathrm{ReLU}(z^{[1]})} \rightarrow \boxed{z^{[2]} = W^{[2]}a^{[1]} + B^{[2]}} \rightarrow \boxed{a^{[2]} = \mathrm{ReLU}(z^{[2]})} \rightarrow \boxed{L(a^{[2]}, y)}$$

$W^{[1]}, B^{[1]}$; $W^{[2]}, B^{[2]}$

$$L(a^{[2]}, y) = (a^{[2]} - y)^2$$

$$\mathrm{ReLU}' = \begin{cases} 1 & (x \geq 0) \\ 0 & (\text{else}) \end{cases}$$

$$da^{[2]} = \frac{dL}{da^{[2]}} = 2(a^{[2]} - y)$$

$$dz^{[2]} = da^{[2]} \cdot \frac{da^{[2]}}{dz^{[2]}} = \begin{cases} 2(a^{[2]} - y) \cdot 1 & (z^{[2]} \geq 0) \\ 0 & \text{otherwise} \end{cases}$$

$$dW^{[2]} = dz^{[2]} \cdot a^{[1]}$$

$$dB^{[2]} = dz^{[2]}$$

$$da^{[1]} = dz^{[2]} \cdot W^{[2]}$$

$$dz^{[1]} = da^{[1]} \cdot \mathrm{ReLU}'(z^{[1]})$$

$$= \begin{cases} da^{[1]} & (z^{[1]} \geq 0,\ z^{[2]} \geq 0) \\ 0 & (\text{otherwise}) \end{cases}$$

$$dW^{[1]} = dz^{[1]} \cdot X$$

$$dB^{[1]} = dz^{[1]}$$

$$W^{[1]} = W^{[1]} - \alpha \cdot dW^{[1]}$$

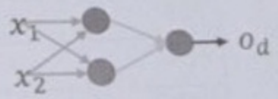
$$W^{[2]} = W^{[2]} - \alpha \cdot dW^{[2]}$$

$$B^{[1]} = B^{[1]} - \alpha \cdot dB^{[1]}$$

$$B^{[2]} = B^{[2]} - \alpha \cdot dB^{[2]}$$

문제 풀어보기.

10. (10점) 아래와 같은 2-layered network 에서 activation function은 ReLu, loss function $E = \sum_{d=1}^{m}(o_d - t_d)^2$ 일 때 (mini batch size = m), $dW^{[1]}, dB^{[1]}, dW^{[2]}, dB^{[2]}$를 계산하고, gradient update rule을 적용하시오.



$$x,\ W^{[1]},\ b^{[1]} \rightarrow \boxed{z^{[1]} = W^{[1]}x + b^{[1]}} \rightarrow \boxed{a^{[1]} = \mathrm{ReLU}(z^{[1]})} \xrightarrow{W^{[2]},\ b^{[2]}} \boxed{z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}} \rightarrow \underbrace{\boxed{a^{[2]} = \mathrm{ReLU}(z^{[2]})}}_{} \rightarrow \boxed{L(a^{[2]}, t_d)}$$

$a^{[2]} = O_d$.

$$da^{[2]} = \frac{dL}{da^{[2]}} = 2\sum_{d=1}^{m}(a^{[2]} - t_d)\cdot 1$$

$$dz^{[2]} = da^{[2]}\cdot\frac{da^{[2]}}{dz^{[2]}} = \begin{cases} 2\sum_{d=1}^{m}(a^{[2]} - t_d)\cdot 1 & (z^{[2]} \geq 0) \\ 0 & \text{otherwise} \end{cases}$$

$$dW^{[2]} = dz^{[2]}\ a^{[1]T} = \begin{cases} 2\sum_{d=1}^{m}(a^{[2]} - t_d)\cdot a^{[1]T} & (z^{[2]} \geq 0) \\ 0 & \text{otherwise} \end{cases}$$

$$db^{[2]} = dz^{[2]}$$

$$da^{[1]} = W^{[2]T}\ dz^{[2]} = \begin{cases} W^{[2]T}\ 2\sum_{d=1}^{m}(a^{[2]} - t_d)\cdot a^{[1]T} & (z^{[2]} \geq 0) \\ 0 & \text{otherwise.} \end{cases}$$

$$dz^{[1]} = da^{[1]} * \underbrace{\mathrm{ReLU}'(z^{[1]})}_{1}$$

$$= \begin{cases} W^{[2]T}\ dz^{[2]}\cdot 1 & (z^{[1]} \geq 0,\ z^{[2]} \geq 0). \\ 0 & \text{otherwise} \end{cases}$$

$$dW^{[1]} = \begin{cases} dz^{[1]}\cdot X^T & (z^{[1]} \geq 0,\ z^{[2]} \geq 0) \\ 0 & \text{otherwise} \end{cases}$$

$$db^{[1]} = \begin{cases} dz^{[1]}. & (z^{[1]} \geq 0,\ z^{[2]} \geq 0) \\ 0 & \text{otherwise} \end{cases}$$

$$W^{[1]} = W^{[1]} - \alpha\, dW^{[1]}$$
$$b^{[1]} = b^{[1]} - \alpha\, db^{[1]}$$
$$W^{[2]} = W^{[2]} - \alpha\, dW^{[2]}$$
$$b^{[2]} = b^{[2]} - \alpha\, db^{[2]}.$$

$$W = W - \alpha \cdot \frac{\quad}{\sqrt{S_{dw}^{corrected}} + \varepsilon}$$

9. (10점)입력 $(x_1, x_2, \cdots, x_n)$, 출력 o_d, loss function이 E일 때, gradient descent with momentum rule을 계산 하시오. (단, 필요한 gradient를 모두 계산하되, 이동평균의 bias correction은 무시해도 좋음)

$$o_d = w_0 + w_1 x_1^{(d)} + \cdots + w_n x_n^{(d)} \qquad E = \sum_{d=1}^{m} (o_d - t_d)^2$$

$$V_{dw} = \beta \cdot V_{dw} + (1-\beta) dw$$

$$W = W - \alpha \cdot V_{dw}$$

$$X, w, b \rightarrow \boxed{a = wx + \underset{(=w_0)}{b}} \rightarrow \boxed{L(a, y)} = (a-y)^2$$

$$da = \frac{dL}{da} = 2(a-y)$$

$$dw = da \cdot X = 2(a-y) \cdot X$$

$$dB = da \cdot 1 = 2(a-y)$$

$$V_{dw} = \beta \cdot V_{dw} + (1-\beta) dw$$

$$V_{dB} = \beta \cdot V_{db} + (1-\beta) db$$

$$W = W - \alpha \cdot V_{dw}$$

$$b = B - \alpha \cdot V_{dB}$$

9. (10점)입력 $(x_1, x_2, \cdots, x_n)$, 출력 o_d, loss function이 E일 때, gradient descent with momentum ?을 계산 하시오. (단, 필요한 gradient를 모두 계산하되, 이동평균의 bias correction은 무시해도 좋음)

V_{dw} 구하는거?

$$o_d = w_0 + w_1 x_1^{(d)} + \cdots + w_n x_n^{(d)}, \quad E = \sum_{d=1}^{m} (o_d - t_d)^2$$

gradient descent

$$V_{dw} := \beta V_{dw} + (1-\beta)\, dW$$
$$V_{db} := \beta V_{db} + (1-\beta)\, db.$$

Initial $V_{dw} = 0.$ $V_{db} = 0.$

$$w := w - \alpha V_{dw}$$
$$b := b - \alpha V_{db}.$$

Let $X = (x_1 \cdots x_n)$

Identity function

$$X \rightarrow \boxed{Z = W^{[1]} X + b^{[1]}.} \rightarrow \boxed{O_d = g(Z^{[1]})} \rightarrow L(O_d, y).$$

$$dO_d = 2 \sum_{d=1}^{m} (O_d - t_d).$$

$$dZ = dO_d \quad \frac{dO_d}{dZ} = 1$$
$$dW = dZ \cdot X^T = 2 \sum_{d=1}^{m} (O_d - t_d) X^T$$
$$db = dZ = 2 \sum_{d=1}^{m} (O_d - t_d)$$

$$V_{dw} = \beta V_{dw} + (1-\beta) \cdot 2 \sum_{d=1}^{m} (O_d - t_d) X^T$$

$$V_{dh} = \beta V_{db} + (1-\beta)\, 2 \sum_{d=1}^{m} (O_d - t_d).$$